



Using Rapise™ with SpiraTest

User Guide

Inflectra Corporation

Date: September 29th, 2011



Contents

Introduction	1
1. Overview	2
2. Configuring SpiraTest	2
3. Connecting Rapise to SpiraTest ...	4
4. Scheduling the Tests	11
5. Configuring RapiseLauncher	15

Introduction

Rapise™ is a next generation software test automation tool that leverages the power of open architecture to improve application quality and reduce time to market.

SpiraTest® provides an integrated, holistic test management solution that manages requirements, tests and incidents in one environment, with complete traceability from inception to completion.

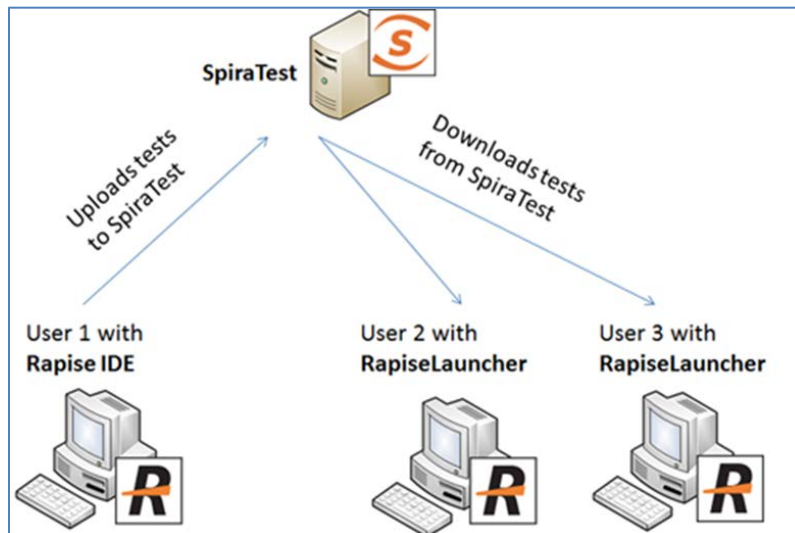
This guide outlines how to use Rapise™ with SpiraTest® (or SpiraTeam®) to centrally manage your automated tests and remotely schedule and launch them in your test lab.

For information regarding how to use SpiraTest itself, please refer to the *SpiraTest User Manual*. For information on using Rapise itself, please refer to the *Rapise User Guide*.

1. Overview

SpiraTest is a web-based quality assurance and test management system with integrated release scheduling and defect tracking. SpiraTest includes the ability to execute manual tests, record the results and log any associated defects.

When you use SpiraTest with Rapise you get the ability to store your Rapise automated tests inside the central SpiraTest repository with full version control and test scheduling capabilities:



You can record and create your test cases using Rapise, upload them to SpiraTest and then schedule the tests to be executed on multiple remote computers to execute the tests immediately or according to a predefined schedule. The results are then reported back to SpiraTest where they are archived as part of the project. Also the test results can be used to update requirements' test coverage and other key metrics in real-time.

Note: SpiraTeam is an integrated ALM Suite that includes SpiraTest as part of its functionality, so wherever you see references to SpiraTest in this section, it applies equally to SpiraTeam.

2. Configuring SpiraTest

Before you can use SpiraTest to manage your Rapise automated tests you need to perform some initial configuration. This section assumes that you already have a working installation of SpiraTest or SpiraTeam v3.1 or later. If not, please refer to the *SpiraTest Installation and Administration Guide* for details on how to install SpiraTest.

2.1. Configuring the Rapise Automation Engine

Log in to SpiraTest as a system administrator and go into SpiraTest main Administration page and click on the "Test Automation" link under **Integration**.

Click the “Add” button to enter the new test automation engine details page. The fields required are as follows:

Edit Engine | New Engine

[<< Back to Test Automation Engine Home](#)

Please enter/edit the following information for the test automation engine. Required fields are indicated in bold:

Name*:

Description:

Token*:

Active

- **Name:** This is the short display name of the automation engine. It can be anything that is meaningful to your users.
- **Description:** This is the long description of the automation engine. It can be anything that is meaningful to your users. (Optional)
- **Active:** If checked, the engine is active and able to be used for any project.
- **Token:** This needs to be the assigned unique token for the automation engine and is used to tell SpiraTest which engine to actually use for a given test case. For Rapise this should always be set to “**Rapise**”.

Once you have finished, click the “Insert & Close” button and you will be taken back to the Automation Engine list page, with Rapise listed as an available automation engine:

Test Automation Engines

SpiraTeam is able to integrate with a variety of external test automation systems using its flexible, open architecture and library of available test automation engines.

This page allows you to view, add and modify the list of test automation engines and make changes to their configuration:

Engine Name	Token	Description	Active	Operations
Apache JMeter	JMeter2		Yes	> Edit Delete
Bad Boy	BadBoy2		Yes	> Edit Delete
Command Line	CommandLine		Yes	> Edit Delete
Rapise 1.3	Rapise	Automation engine for Inflectra Rapise	Yes	> Edit Delete
Selenium	Selenium	Engine that integrates with the open-source Selenium RemoteControl (RC)	Yes	> Edit Delete

2.2. Entering the Test Cases

Open up a project in SpiraTest and go to the Testing > Test Cases tab to display the list of test cases in the project:

Test Case Name	Execution Status	Owner	Last Executed	Author	Active	Test #	Edit
Functional Tests (5)	Passed	Fred Bloggs	9-Sep-2011	Fred Bloggs	Yes	TC000001	Edit
Ability to create new book	Failed	Fred Bloggs	20-Sep-2011	Fred Bloggs	Yes	TC000002	Edit
Ability to edit existing book	Failed	Fred Bloggs	21-Sep-2011	Fred Bloggs	Yes	TC000003	Edit
Ability to create new author	Passed	Joe P Smith	9-Sep-2011	Fred Bloggs	Yes	TC000004	Edit
Ability to edit existing author	Failed	Joe P Smith	20-Sep-2011	Fred Bloggs	Yes	TC000005	Edit
Ability to reassign book to different author	Passed	Joe P Smith	20-Sep-2011	Fred Bloggs	Yes	TC000006	Edit
Regression Tests (2)	Passed	Joe P Smith	9/20/2011 2:39:02 PM	Joe P Smith	Yes	TC000007	Edit
Book management	Passed	Joe P Smith	1-Dec-2003	Joe P Smith	Yes	TC000008	Edit
Author management	Passed	Joe P Smith	1-Dec-2003	Joe P Smith	Yes	TC000009	Edit
Scenario Tests (2)	Passed	Joe P Smith	1-Dec-2003	Joe P Smith	Yes	TC000010	Edit
Common Tests (2)	Passed	Fred Bloggs	1-Dec-2003	Fred Bloggs	Yes	TC000015	Edit
Open Up Web Browser	Passed	Fred Bloggs	1-Dec-2003	Fred Bloggs	Yes	TC000016	Edit
Login to Application	Passed	Fred Bloggs	1-Dec-2003	Fred Bloggs	Yes	TC000017	Edit

You can either create a new test case that will be used to store the Rapise automated test or simply use an existing test case. In this example we shall create a new test case called “Create New Book - Automated”. Once that’s added, it will appear in the list with status “Not Run”:

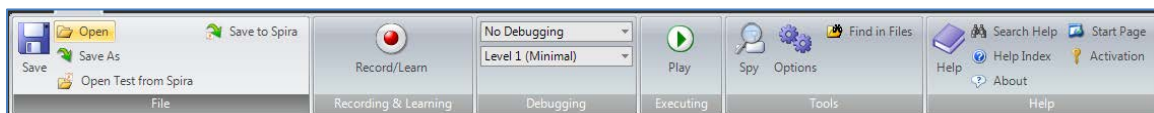
Test Case Name	Execution Status	Owner	Last Executed	Author	Active	Test #	Edit
Functional Tests (6)	Passed	Fred Bloggs	9-Sep-2011	Fred Bloggs	Yes	TC000001	Edit
Create New Book - Automated	Not Run	System Administrator		System Administrator	Yes	TC000041	Edit
Ability to create new book	Failed	Fred Bloggs	20-Sep-2011	Fred Bloggs	Yes	TC000002	Edit
Ability to edit existing book	Failed	Fred Bloggs	21-Sep-2011	Fred Bloggs	Yes	TC000003	Edit
Ability to create new author	Passed	Joe P Smith	9-Sep-2011	Fred Bloggs	Yes	TC000004	Edit
Ability to edit existing author	Failed	Joe P Smith	20-Sep-2011	Fred Bloggs	Yes	TC000005	Edit
Ability to reassign book to different author	Passed	Joe P Smith	20-Sep-2011	Fred Bloggs	Yes	TC000006	Edit
Regression Tests (2)	Passed	Joe P Smith	1-Dec-2003	Joe P Smith	Yes	TC000007	Edit
Book management	Passed	Joe P Smith	1-Dec-2003	Joe P Smith	Yes	TC000008	Edit
Author management	Passed	Joe P Smith	1-Dec-2003	Joe P Smith	Yes	TC000009	Edit
Scenario Tests (2)	Passed	Joe P Smith	1-Dec-2003	Joe P Smith	Yes	TC000010	Edit
Common Tests (2)	Passed	Fred Bloggs	1-Dec-2003	Fred Bloggs	Yes	TC000015	Edit
Open Up Web Browser	Passed	Fred Bloggs	1-Dec-2003	Fred Bloggs	Yes	TC000016	Edit
Login to Application	Passed	Fred Bloggs	1-Dec-2003	Fred Bloggs	Yes	TC000017	Edit

Now that we have created the test case, the next step is to write the Rapise automated test and upload it into SpiraTest, linked to this particular test case

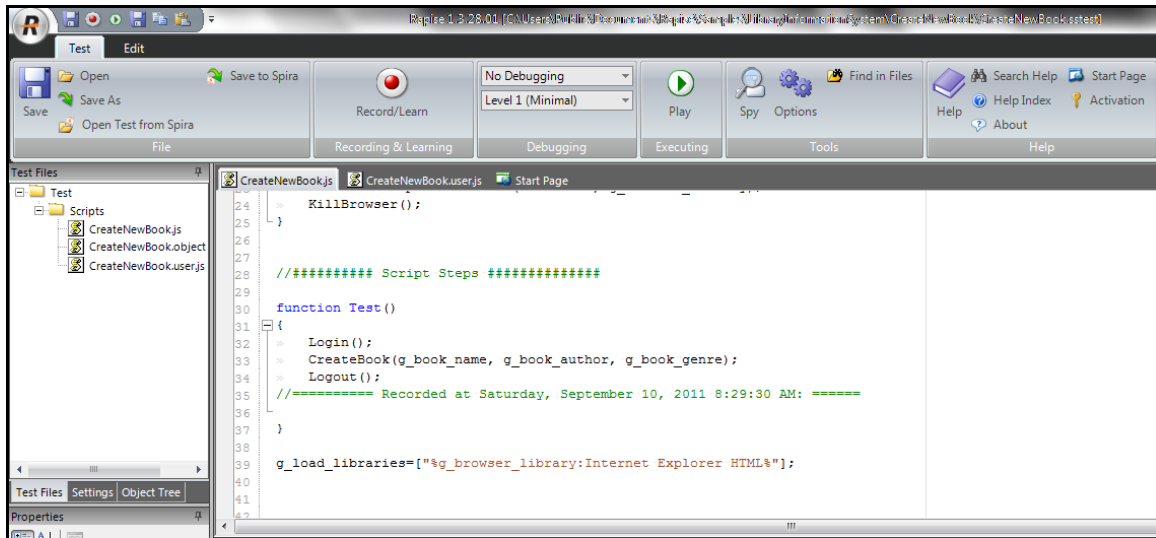
3. Connecting Rapise to SpiraTest

The next step is to open Rapise on the workstation that will be used to record/create the automated test. You can either choose the option to create a new test from scratch or simply open an existing test. In either case, we can upload the test to SpiraTest and link it to this test case. For this example we shall open one of the existing sample test cases that come with SpiraTest.

Within Rapise, click on the Open button in the File section of the Rapise Test ribbon:



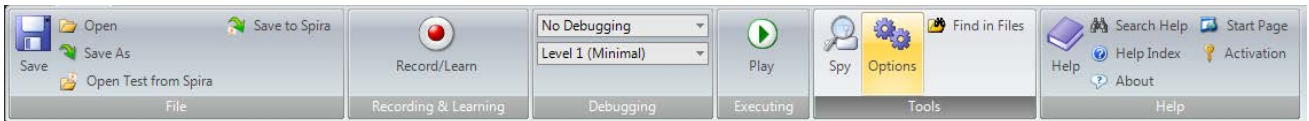
Then navigate to the `C:\Users\Public\Documents\Rapise\Samples\Library\InformationSystem` folder and choose to open the `CreateNewBook.sstest` test. This test will open inside Rapise:



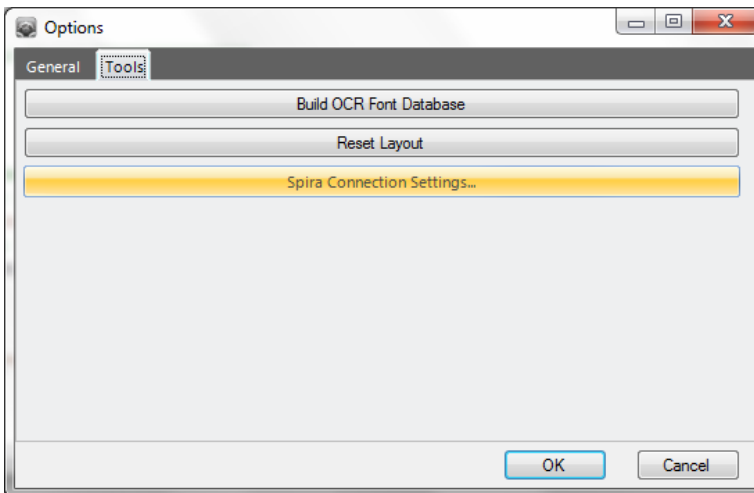
Before saving the test to SpiraTest, click on the Play button to verify that the test works. It should launch the sample Library Information System web application and create a new book in the system.

3.1. Configuring the SpiraTest Connection

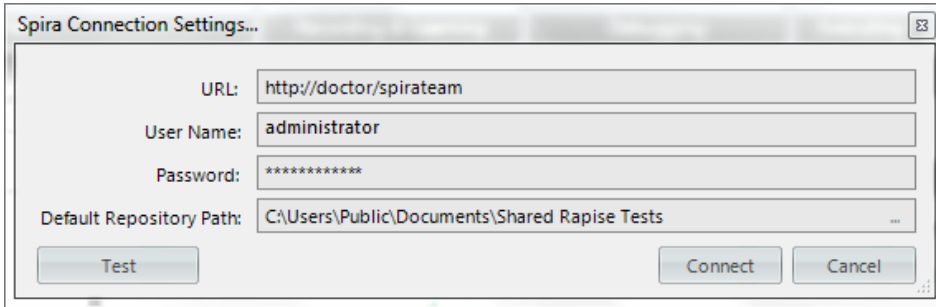
Once you have verified that the test works correctly, the next thing you need to do is to configure the connection to SpiraTest. To do this, click on the Options button in the Tools section of the Rapise Test ribbon:



This will bring up the Options dialog box. Click on the Tools tab to bring up the settings related to the various Tools:



Click on the "Spira Connection Settings" button to bring up the dialog box that lets you configure the connection to SpiraTest:

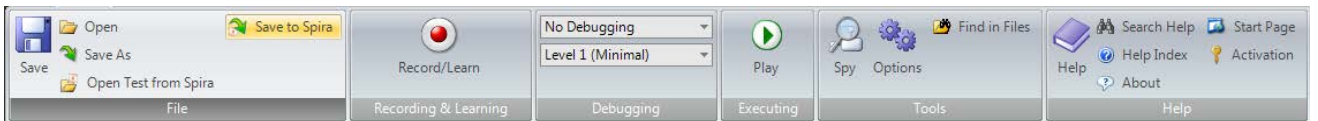


Enter the URL, login and password that you use to connect to SpiraTest and then click the "Test" button to verify that the connection information is correct.

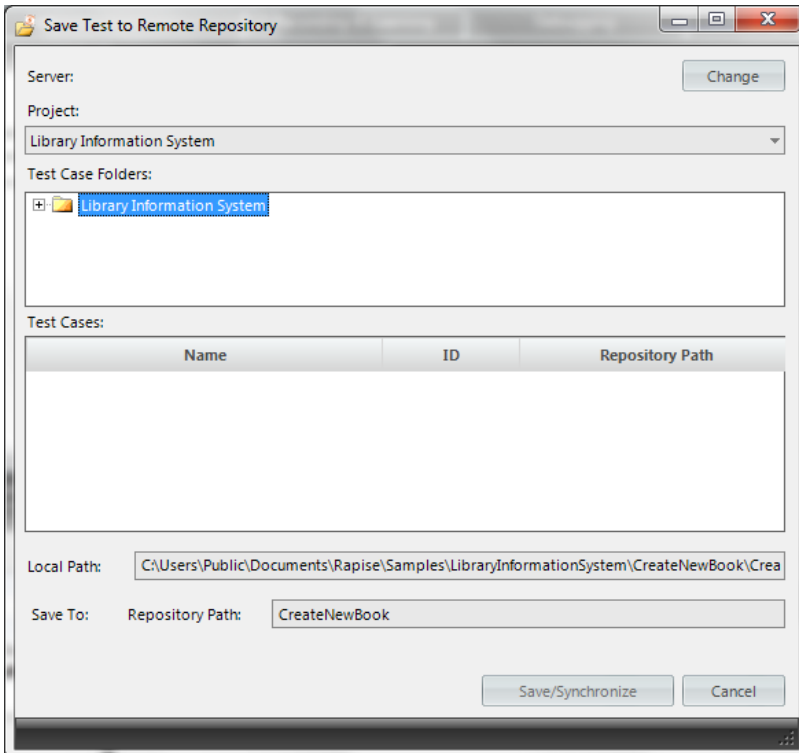
Note that the "Default Repository Path" is a folder that used to store local copies of the non-absolute repositories.

3.2. Saving the Rapise Test to SpiraTest

Now, click on the **Save to Spira** icon in the File section of the Rapise Test ribbon:

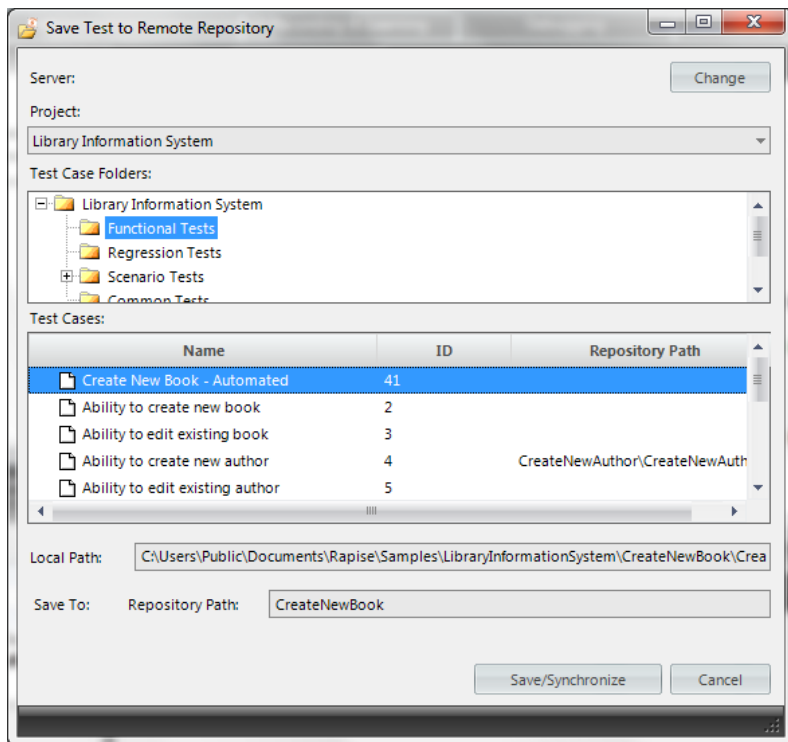


That will bring up the Save to SpiraTest dialog box:

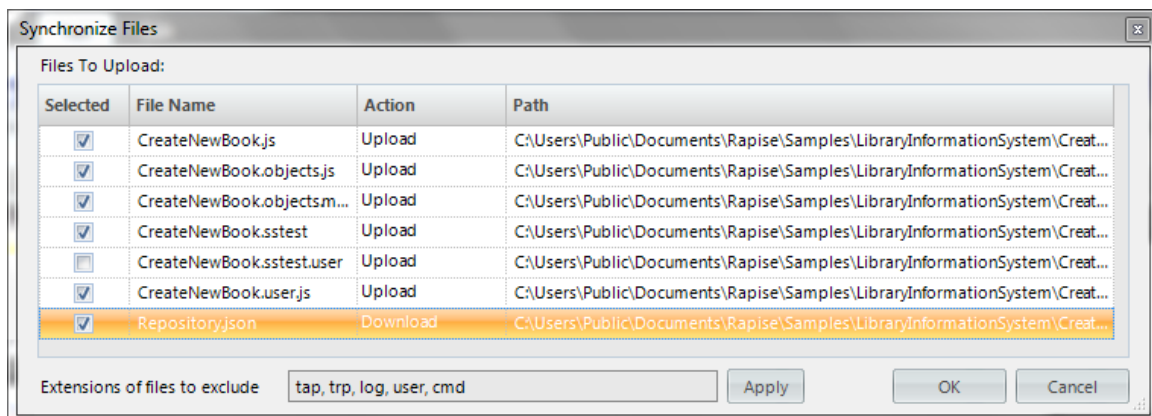


The first thing you will need to do is choose the SpiraTest project from the dropdown list. Once you have done that, the system will ask you to choose the local Repository path for the Rapise test. This is when the test will get saved-to locally before checking-in to from SpiraTest. Also it's where the test gets checked-out to if you open the test in the future.

Once you have chosen the repository path, you need to expand the test case folders in SpiraTest and choose the existing Test Case that you want to attach the Rapise test to:



When you expand the folders to display the list of contained test cases, it will display the name of the associated Rapise test script associated with it (to the right). In this case, please choose the “Create New Book – Automated” test case that was just created. Then click the [Save/Synchronize] button:



A dialog box will be displayed that lists all the files in the local working directory and shows which ones will be checked-in to SpiraTest. The system will filter out result and report files that shouldn't be uploaded. You can change which files are filtered out and also selectively include/exclude files. Once you are happy with the list of files being checked-in, click the [OK] button.

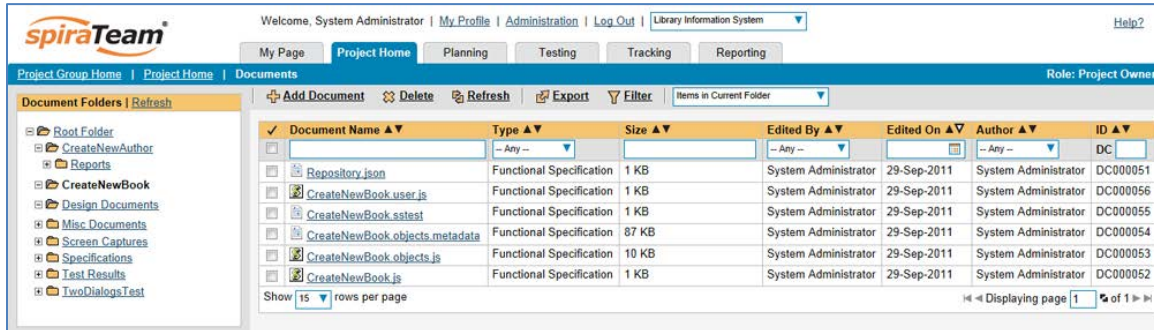
If an error occurs during the save, a message box will be displayed, otherwise the dialog box will simply close. You now have successfully uploaded the Rapise test to SpiraTest. If you make changes, you can synchronize those changes with SpiraTest. SpiraTest has a built-in version control system and will keep a list of all the revisions made to the test.

Synchronizing

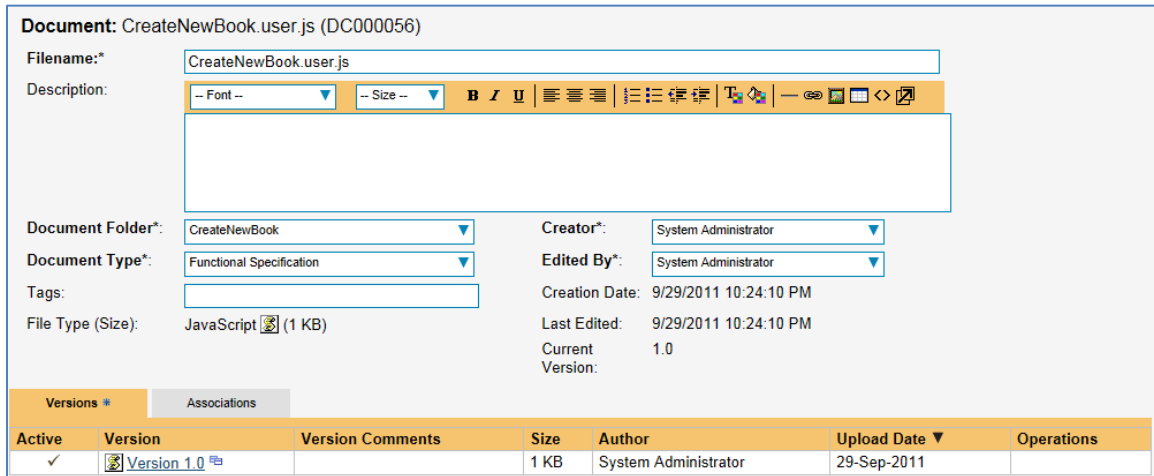
Note: When you have a Rapise test previously loaded from SpiraTest or saved to SpiraTest then the standard "Open" and "Save" buttons should not be used. Suppose for example, you have some files modified locally and want to update them in the SpiraTest repository. When you click on the Save/Synchronize button, Rapise will show you which files have been modified locally and which have been modified in the central repository. The synchronization dialog either allows you to upload/download the revisions or keep intact the locally modified files depending on your choice.

3.3. Viewing the SpiraTest Repository

If you open up the project in SpiraTest and click on the Project Home > Documents link, you will be taken to the central document repository that now includes your new Rapise test folders:



To see the different revisions of a file, simply click on the hyperlink for a repository item and the list of previous revisions will be displayed:



If you click on the Testing > Test Cases tab and then click on the test case that we previously created you will see that it now has its Automation information populated:

Test Case: Create New Book - Automated [TC:000041]

Name*: Create New Book - Automated

Description:

Author*: System Administrator **Est. Dur.:** [] hours

Owner: -- None -- **Creation Date:** 9/29/2011 9:24:32 PM

Priority: -- None -- **Execution Status:** [Not Run](#)

Active*: Yes **Last Executed:**

Test Steps Req. Coverage **Automation *** Comments Custom Props * Test Runs Releases

This section defines the automated test script associated with this test case:

Automation Engine*: Rapise 1.3

Script Type*: Attached Linked

Filename*: CreateNewBook\CreateNewBook.sstest

Document Type*: Functional Specification

Document Folder*: Root Folder

Version: v 1.0

This shows you that the “Create New Book – Automated” test case is now linked to the corresponding Rapise test stored in the SpiraTest document repository (in the CreateNewBook folder). If for any reason you move the CreateNewBook folder to a different subfolder in the document repository, you should update the filename listed in the test case “Filename” field.

3.4. Using Parameterized Test Cases

Often you will have an automated test script that you want to run several times using:

- Different browsers (e.g. Firefox, Chrome and Internet Explorer)
- Different test data

You can define the various test parameters for both these cases and have SpiraTest pass the values through to the Rapise automated test. For example, on the Automation tab of the new test case, click the “Edit Parameters” link and enter the following information:

Edit Test Case Parameters [X]

The following parameters have been defined for this test case:

Name	Default Value	Operations
\$_g_book_author	null	Copy To Clipboard Delete
\$_g_book_genre	null	Copy To Clipboard Delete
\$_g_book_name	null	Copy To Clipboard Delete

Add a new parameter to this test case:

Name*:

Default Value*:

[Add](#) [Cancel](#)

We have defined four input parameters for this test case:

- **g_book_author** – This contains the name of the author for the new book being created
- **g_book_genre** – This contains the name of the genre for the new book being created
- **g_book_name** – This contains the title/name of the new book being created
- **g_browser_library** – This contains the name of the browser we should use to run the test

Now for these parameters to actually affect the Rapise test, you need to make sure that the Rapise test is expecting these variables and knows how to handle them. Conventionally in Rapise, all **global variables** are prefixed with **g_** which is why we have similarly named the SpiraTest parameter names. In the sample Library Information System test, we have the following code:

```
function TestInit()
{
    //Input variables - allows SpiraTest to configure as parameters
    //Provide backup values if not defined
    if ('undefined' == typeof(g_book_name))
    {
        g_book_name = 'The Restaurant at the end of the Universe';
    }
    if ('undefined' == typeof(g_book_author))
    {
        g_book_author = 'Agatha Christie';
    }
    if ('undefined' == typeof(g_book_genre))
    {
        g_book_genre = 'Science Fiction';
    }
    if ('undefined' == typeof(g_browser_library))
    {
        g_browser_library = "Internet Explorer HTML";
    }

    Tester.SetReportAttribute("Browser", g_browser_library);
    KillBrowser();
}
```

This code will check to see if the variables are provided by SpiraTest and if not, it will use some defaults. This is useful when you want to be able to run the test directly from Rapise and from SpiraTest without having to make changes to the test script.

These global variables (`g_book_name`, `g_book_author`, `g_book_genre`) can now be used in the Rapise test script at the appropriate points in the playback.

The `g_browser_library` variable is used to specify which browser should run the test. This is done with the following command, located in the `CreateNewBook.js` file:

```
g_load_libraries=["%g_browser_library:Internet Explorer HTML%"];
```

This tells Rapise to use either the `g_browser_library` variable (if defined) or fallback to using Internet Explorer if not.

Note: Parameters in SpiraTest are always lower case, so any Rapise variables that will be populated by SpiraTest need to also be in lower-case only.

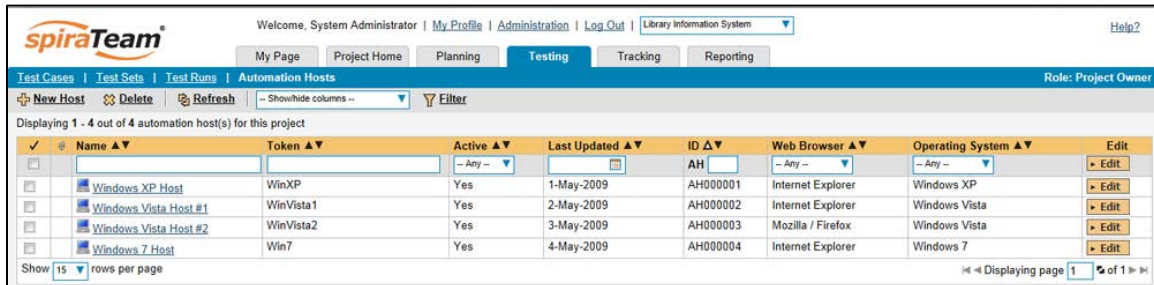
4. Scheduling the Tests

Now that we have our Rapise automated test uploaded to SpiraTest and associated with a test case, we can now schedule the test to be executed. The first thing we need to do is setup the list of automation hosts.

4.1. Configuring the Automation Hosts

When you execute Rapise automated tests from SpiraTest, you have the ability to specify which computer(s) it will be executed on. We call those different computers, the “automated hosts”. Each automation host needs to have a copy of Rapise installed on it.

Go to Testing > Automation Hosts in SpiraTest to display the list of automation hosts:



The screenshot shows the SpiraTest interface for the 'Automation Hosts' section. The page title is 'Automation Hosts' and the role is 'Project Owner'. There are navigation tabs for 'My Page', 'Project Home', 'Planning', 'Testing', 'Tracking', and 'Reporting'. The 'Testing' tab is active. Below the navigation, there are buttons for 'New Host', 'Delete', and 'Refresh'. A filter is applied, showing '1 - 4 out of 4 automation host(s) for this project'. The table below lists the automation hosts with columns for Name, Token, Active, Last Updated, ID, Web Browser, and Operating System.

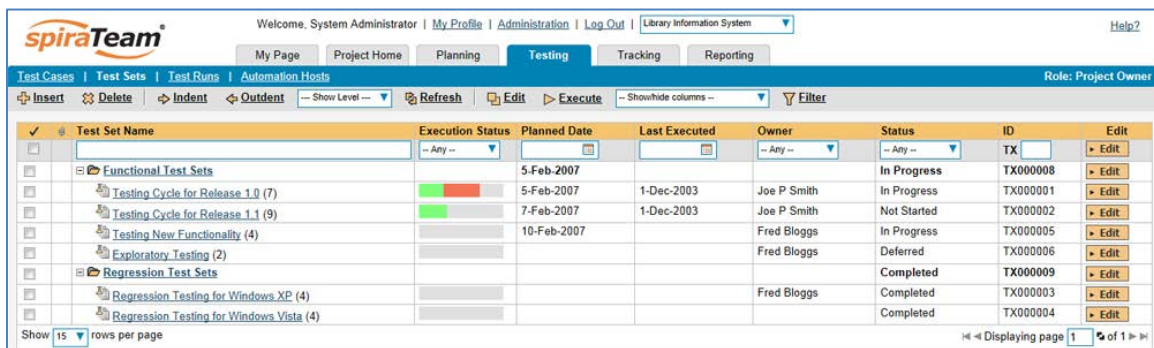
Name	Token	Active	Last Updated	ID	Web Browser	Operating System	Edit
Windows XP_Host	WinXP	Yes	1-May-2009	AH000001	Internet Explorer	Windows XP	Edit
Windows Vista_Host #1	WinVista1	Yes	2-May-2009	AH000002	Internet Explorer	Windows Vista	Edit
Windows Vista_Host #2	WinVista2	Yes	3-May-2009	AH000003	Mozilla / Firefox	Windows Vista	Edit
Windows 7_Host	Win7	Yes	4-May-2009	AH000004	Internet Explorer	Windows 7	Edit

Make sure that you have created an Automation Host for each computer that is going to run an automated test case. The name and description can be set to anything you like, but the Token field must be set to a **unique name for each computer**.

Once you have at least one Automation Host configured, we need to next create the test sets that will be scheduled to execute on these hosts.

4.2. Creating and Scheduling the Test Sets

Inside SpiraTest, click on Testing > Test Sets to display the list of existing test sets:



The screenshot shows the SpiraTest interface for the 'Test Sets' section. The page title is 'Test Sets' and the role is 'Project Owner'. There are navigation tabs for 'My Page', 'Project Home', 'Planning', 'Testing', 'Tracking', and 'Reporting'. The 'Testing' tab is active. Below the navigation, there are buttons for 'Insert', 'Delete', 'Indent', 'Outdent', 'Refresh', 'Edit', and 'Execute'. A filter is applied, showing '1 - 4 out of 4 test set(s) for this project'. The table below lists the test sets with columns for Test Set Name, Execution Status, Planned Date, Last Executed, Owner, Status, and ID.

Test Set Name	Execution Status	Planned Date	Last Executed	Owner	Status	ID	Edit
Functional Test Sets	In Progress	5-Feb-2007			In Progress	TX000008	Edit
Testing Cycle for Release 1.0 (7)	In Progress	5-Feb-2007	1-Dec-2003	Joe P Smith	In Progress	TX000001	Edit
Testing Cycle for Release 1.1 (9)	In Progress	7-Feb-2007	1-Dec-2003	Joe P Smith	Not Started	TX000002	Edit
Testing New Functionality (4)	In Progress	10-Feb-2007		Fred Bloggs	In Progress	TX000005	Edit
Exploratory Testing (2)	Deferred			Fred Bloggs	Deferred	TX000006	Edit
Regression Test Sets	Completed				Completed	TX000009	Edit
Regression Testing for Windows XP (4)	Completed			Fred Bloggs	Completed	TX000003	Edit
Regression Testing for Windows Vista (4)	Completed				Completed	TX000004	Edit

Now click on **Insert** to create a new test set that will contain our automated test case:

Test Set Name	Execution Status	Planned Date	Last Executed	Owner	Status	ID	Edit
Functional Test Sets		5-Feb-2007			In Progress	TX000008	Edit
Rapise Automated Tests					Not Started		Insert Update Cancel
Testing Cycle for Release 1.0 (7)		5-Feb-2007	1-Dec-2003	Joe P Smith	In Progress	TX000001	Edit
Testing Cycle for Release 1.1 (9)		7-Feb-2007	1-Dec-2003	Joe P Smith	Not Started	TX000002	Edit
Testing New Functionality (4)		10-Feb-2007		Fred Bloggs	In Progress	TX000005	Edit
Exploratory Testing (2)				Fred Bloggs	Deferred	TX000006	Edit
Regression Test Sets					Completed	TX000009	Edit
Regression Testing for Windows XP (4)				Fred Bloggs	Completed	TX000003	Edit
Regression Testing for Windows Vista (4)					Completed	TX000004	Edit

Note: Unlike manual test cases, automated test cases must be executed within a test set – they cannot be executed directly from the test case.

Now click on the hyperlink of our new test set to display the test set details page:

Test Set: Rapise Automated Tests [TX:000015]

Name*: Rapise Automated Tests

Description: [Rich Text Editor]

Owner: [None] Creator*: System Administrator

Release: [None] Type*: Manual

Automation Host: [None] Creation Date: 9/30/2011 9:14:39 AM

Status*: Not Started Last Executed: -

Planned Date: [] Last Updated: 9/30/2011 9:16:05 AM

Test Cases | Test Runs | Comments | Custom Props # | Attachments | History #

> Add Tests | Remove Tests | Refresh | Edit Parameters | Execute Tests Est. Dur.: 0.0h / Act. Dur.: 0.0h

Test Case Name	Owner	Priority	Est. Dur.	Act. Dur.	Last Executed	Execution Status	ID	Edit
Show 15 rows per page								

First, click on the “Add Tests” hyperlink in the “Test Cases” tab to bring up the dialog box that lets us add new test cases to the set:

Add Test Case to the Test Set

Choose the test case(s) to add to the current test set:

- Test Case
- Functional Tests
- Create New Book - Automated
- Ability to create new book
- Ability to edit existing book
- Ability to create new author
- Ability to edit existing author

> Add | Cancel

Choose our new Rapise automated test case and click the ‘Add’ button. The test case will now be added to the set. Perform the same action again so that we have the separate instances of our Rapise test case:

status of the test set to “In Progress”, and once test execution is done, the status of the test set will change to either “Completed” – the automation engine could be launched and the test has completed – or “Blocked” – RapiseLauncher was not able to execute the automation test.

If you want to immediately execute the test case on your local computer, instead of setting the “Automation Host”, “Status” and “Planned Date” fields, you can instead click the [Execute] icon on the test set itself. This will cause RapiseLauncher on the local computer to immediately start executing the current test set.

In either case, once all the test cases in the test set have been completed, the status of the test set will switch to “Completed” and the individual test cases in the set will display a status based on the results of the Rapise test:

- **Passed** – The Rapise automated test ran successfully and all the test conditions in the test script passed
- **Failed** – The Rapise automated test ran successfully, but at least one test condition in the test script failed.
- **Blocked** – The Rapise automated test did not run successfully

If you receive the “Blocked” status for either the test set or the test cases you should open up the Windows Application Event Log on the computer running RapiseLauncher and look in the event log for error messages.

Note: While the tests are executing you may see browser or application windows launch as Rapise executes the appropriate tests.

Once the tests have completed, you can log back into SpiraTest and see the execution status of your test cases. If you click on a Test Run that was generated by Rapise, you will see the following information:

Test Run: Create New Book - Automated [TR:000190]

Release #: -- None -- Estimated Duration: [] hours
Tester Name:* System Administrator Actual Duration:* 0.0 hours
Test Set: Rapise Automated Tests Execution Date: 10/3/2011 1:07:19 PM
Test Case #:* TC000041 Execution Status:* **Failed**
Automation Host: Windows 7 Host Test Run Type:* Automated

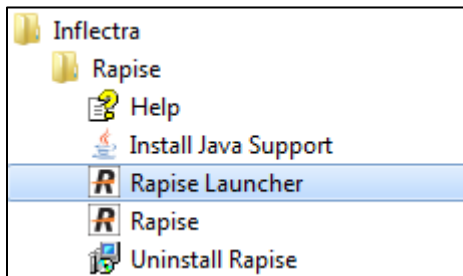
Test Run Steps Automation * Custom Props * Attachments

Runner Name: Rapise Assert Count: 0
Message: 11 Steps Passed, 2 Steps Failed (84% Success Rate) Test Name: CreateNewBook

Details:
Script Execution Report
=====
CreateNewBook
Process 'chrome.exe' was not terminated
ok 1 - Chrome Process Created
not ok 2 - Connected to Chrome window
Failed test
ok 3 - Log In.DoClick()
ok 4 - Username: DoSetText(["librarian"])
ok 5 - Password: DoSetText(["librarian"])
ok 6 - ctI00\$MainContent\$LoginUser\$LoginButton.DoClick()
ok 7 - Book Management.DoClick()
ok 8 - (Create new book) .DoClick()
ok 9 - Name: DoSetText(["Bleak House"])
ok 10 - Author: DoSelect(["Charles Dickens"])
ok 11 - Genre: DoSelect(["Historical Fiction"])
ok 12 - ctI00\$MainContent\$btnSubmit.DoClick()
not ok 13 - Failure in Test
Failed test

5. Configuring RapiseLauncher

RapiseLauncher is an add-on for Rapise that installs along with the main Rapise application. It can be found in the Start > Programs > Inflectra > Rapise program folder:



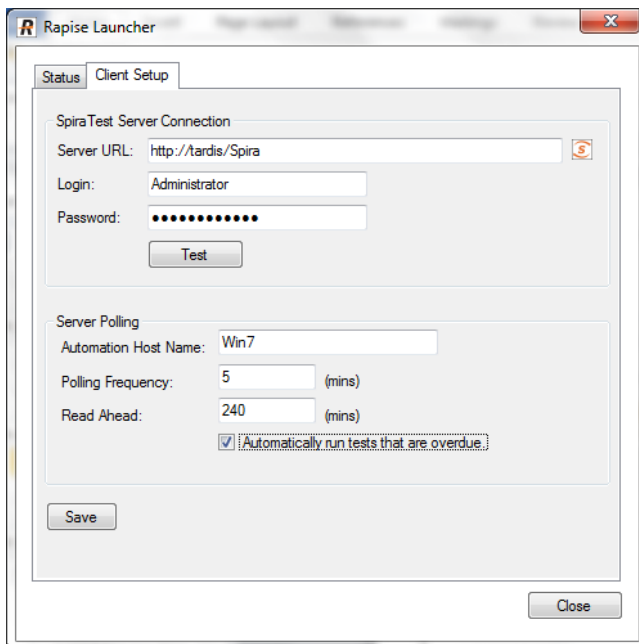
5.1 Basic Unattended Operation

When run, the program will start minimized to the system tray and will start its polling of the server. Polling will occur every 'x' minutes (60 by default) for any automated test sets that are scheduled to be run. When time comes for a test to be launched, it will start Rapise to execute the test. Rapise will then perform the test activities and report the results back to SpiraTest.

At the end of the test, the program will go back and resume scanning for tests that need to be executed. Typically (unless there is a bug in the test or application being tested) no user input is ever needed from the application itself.

5.2. Client Configuration


By right clicking on the system tray icon and selecting "Configuration", the application's window will open to the configuration panel.



The panel has the following options:

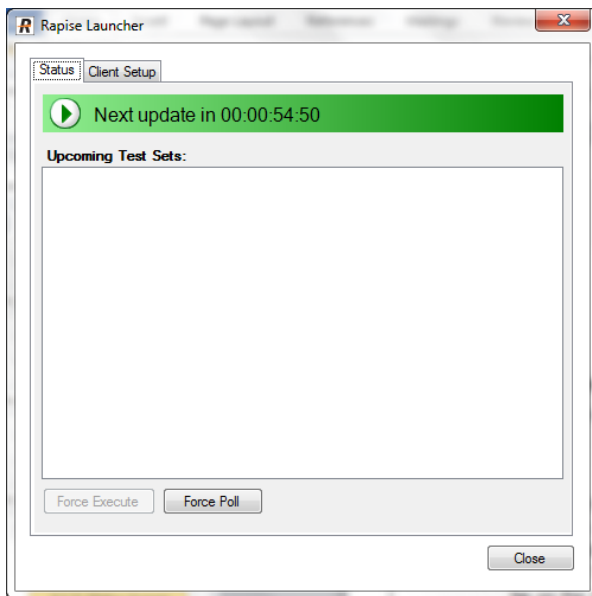
- SpiraTest Server Configuration:
 - **Server URL:** This is the URL of the SpiraTest installation. Be sure to not put /Login.aspx or any other page in the string, this should be just the root URL of the application's install.

- **Login Username:** This is the SpiraTest login id of the user that you want the tests reported as. Note that while the application is polling and updating test results, if the user is logged into a web browser session, they will get kicked out.
- **Login Password:** The password to the Username above.
- **Test:** Clicking this will test the login to make sure the application can connect to the server properly.
- **Server Polling:**
 - **Automation Host Name:** This field is required, and uniquely identifies the local testing machine. Any scheduled tests assigned to the Automation Host on SpiraTest will get polled for this machine. Except in special circumstances, this ID should be unique among all testing machines.

 **Important:** This field must match the string that is entered into the Automation Host Details screen in the **Token:** field, or scheduled tests will not be recognized.
 - **Automatically Run Overdue Tests:** When this is checked, any tests that are pulled from the SpiraTest server that has a scheduled date in the past will be marked as Overdue. Normally, overdue tests will not be executed. With this check, they will be executed as soon as the poll is finished.
 - **Polling Frequency:** How often in minutes the application will poll the SpiraTest server for updates to the automation host's schedule. The default is 60 (1 hour), and should be fine for most installations. Note that tests will still be executed on their scheduled time, this is simply how often the program will talk to the SpiraTest server to detect schedule changes. Updating the polling frequency will reset the currently running timers.
 - **Polling Read Ahead:** How far ahead in minutes the program should read the schedule for the Automation host. Tests that are scheduled farther in advance will not show up as a pending test on the status screen.

5.3. Status Screen

The status screen is usually hidden, but can be brought up for display by double-clicking on the system tray icon:



The top of the screen shows the current status, whether it's running a test or waiting to poll the server for an update. It will also show any errors present on the application, like a registration error or configuration issue. Under the status bar is a list of any pending or executing tests that are scheduled for this testing machine. The list will get cleared at every poll, so tests that have executed since the previous poll will still be on the list, and will show their execution status:

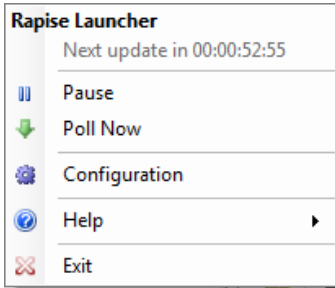
- **Green Arrow:** A green arrow indicates that the test is still running, or RapiseLauncher is waiting for a reply from Rapise.
- **Blue Checkbox:** A blue checkbox indicates that the test is completed, regardless of status of the individual test steps in the scheduled test set.
- **Red Error:** A red error indicator indicates that Rapise ran into an issue (outside of test results). In this case, any further tests will be marked as blocked, as the issue needs to be corrected within Rapise or the Rapise test script.
- **No Indication:** No indication means that the test is currently awaiting for its scheduled date to start. Note that only one test will be launched at a time, so that if two tests are scheduled at the same time, the one with the lower TestSet ID will be executed first, then as soon as it's finished, the second scheduled test will be run.

By highlighting a test that has not been executed yet, you can click the **Force Execute** button. This will cause the selected test to have its scheduled date to the current time, causing it to be immediately executed (or, if another test is already running, next in line for execution).

At any time the **Force Poll** button can be clicked, causing RapiseLauncher to initiate an immediate poll of the SpiraTest server to check for pending runs. The timers for the next server poll will be reset when the button is clicked.

5.4. Tray Icon Menu

Instead of operating from the application window, all functions exist on the tray icon menu as well, as well as some additional commands:



- **Pause / Resume:** The Pause/Resume option pauses or resumes the timers for polling and executing tests. If a test or server poll is already in progress, it will not cancel these. However, after they are finished, no further polls or tests will be run.
- **Poll Now:** This will force a server poll for upcoming tests, and reset the poll timer.
- **Configuration:** Opens the main window to the Configuration page.
- **Help -> About:** Opens the About window, which displays information about Rapise Launcher.
- **Help -> View Help:** Opens this PDF file in a browser.
- **Exit:** Will completely exit the program. Doing this will cancel any tests currently running and shut down the program. Any tests that were waiting to be executed will not execute until the program is restarted and the polling is resumed.

You can double-click the tray icon to bring up the main window on the Status page.

5.5. Running RapiseLauncher from a Build Script

Normally you schedule tests in SpiraTest using the Planned Date field of the test sets and let the various instances of RapiseLauncher poll SpiraTest for upcoming tests. In addition (as described in the *SpiraTest User Manual*) you can execute a test set on the local machine immediately by clicking the “Execute” button within SpiraTeam.

However there are situations where you want to be able to launch an automated Rapise test script from an external batch file or build script (e.g. as part of a continuous integration environment) and have those tests report their results back into SpiraTest. You can achieve this by using the special command-line argument `-testset` which is passed to RapiseLauncher. For more details on this parameter see the next section.

5.6. Command line arguments

For debugging and additional options when running the program, the following command-line arguments are available:

-status	Shows the Status screen upon startup. (Normal action is to run minimized to the system tray.)
-paused	Starts the application with timers Paused instead of active.
-poll	Forces the program to do an initial poll upon startup. (Normal action is to wait the pending time before doing the initial poll.)
-trace	Enables tracelogging to the EventLog for debugging and watching tests execute.

-logfile	Forces events to be written to a text file instead of the Application EventLog. This option enables -trace as well. Files are located in the Local Application Data folder. (C:\Users\ <user>\AppData\Local on Vista/Win7).</user>
-testset:[Test Set ID]	Allows you to tell RapiseLauncher to execute a specific test set on the remote computer (e.g. -testset:45 runs test set TX00045)
<filename>	Must be the last item on the command line. This is a TST file downloaded from SpiraTest to start immediate execution on.

Legal Notices

This publication is provided as is without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

This publication could include technical inaccuracies or typographical errors. Changes are periodically added to the information contained herein; these changes will be incorporated in new editions of the publication. Inflectra Corporation may make improvements and/or changes in the product(s) and/or program(s) and/or service(s) described in this publication at any time.

The sections in this guide that discuss internet web security are provided as suggestions and guidelines. Internet security is constantly evolving field, and our suggestions are no substitute for an up-to-date understanding of the vulnerabilities inherent in deploying internet or web applications, and Inflectra cannot be held liable for any losses due to breaches of security, compromise of data or other cyber-attacks that may result from following our recommendations.

SpiraTest[®], SpiraPlan[®], SpiraTeam[®], Rapise[™] and Inflectra[®] are either trademarks or registered trademarks of Inflectra Corporation in the United States of America and other countries. Microsoft[®], Windows[®], Explorer[®] and Microsoft Project[®] are registered trademarks of Microsoft Corporation. All other trademarks and product names are property of their respective holders.

Please send comments and questions to:

Technical Publications

Inflectra Corporation

8121 Georgia Ave, Suite 504

Silver Spring, MD 20910-4957

U.S.A.

support@inflectra.com