



# SMARTBEAR

## Test Automation Integration with Test Management QAComplete

This User's Guide walks you through configuring and using your automated tests with  
QAComplete's Test Management module

SmartBear Software

Release 11

October 2015

## About Automated Tests

Test Management's Automation feature allows you to run tests with little to no human interaction. You can perform testing from remote computers as often as needed - for example, after a build or on a daily schedule.

By using the QAComplete integration with various testing tools, you can fully track tests and test results across Test Management's manual and automated tests. You can run automated tests from the desired computers and post results for full tracking of your testing coverage and progress.

### Supported Automated Testing Tools

- SmartBear's TestComplete. For more information about TestComplete, please see [TestComplete by SmartBear](#).
- SmartBear's Ready! API. For more information about Ready! API, please see [Ready! API by SmartBear](#).
- SmartBear's SoapUI. For more information on SoapUI, please see [SoapUI by SmartBear](#).
- Selenium WebDriver for the following unit testing frameworks:
  - JUnit
  - TestNG
  - NUnit

For more information about Selenium, please see [Selenium - Web Browser Automation](#).

You can also automate the running of ordinary JUnit and NUnit unit tests (that do not use Selenium modules).

**Note:** Starting with version 10.50, **TestComplete** supports integration with Selenium tests. In TestComplete, you can run Selenium tests created by using **JUnit**, **TestNG**, **NUnit**, **PyUnit**, **Ruby Test::Unit** and **PHPUnit** unit testing frameworks.

You can add your Selenium tests to your TestComplete project, and then integrate the project with Test Management.

### How Test Automation Works

- In your automated testing tool, create automated tests that will perform the needed actions to test your application or feature.
- Pack your automated test files into a .zip archive. You can store the zipped test on a shared network folder or web site. You can also upload the files directly to your Tests in the Test Library.
- Install and configure Test Agent on the test lab computers that will run your automated tests.

- In Test Management, associate your automated tests with Tests in the Test Library, whether pointing to the shared drive or uploading directly to the Test Library tests.
- In Test Management, you can also create Automation Schedules to run the automated tests on a desired schedule, or submit the Tests to run on demand.
- When the Automated Tests run, the Test Agent on the Test Host:
  - Connects to the QAComplete web service.
  - Gets automated tests assigned to the test.
  - Starts the needed automated testing tool on the Test Host.
  - Runs the test.

**Important:** The Test Agent can be set to log off of Windows on the Host machine. This effectively closes any open applications, logs out, then logs back in with the credentials from the Test Agent configuration. There is an option on the configuration form to first check if that user is currently logged in, and avoid the log out process.

- After the test run is complete, results are posted to the Test and/or Test Sets in Test Management.

You can review the results on the Test Management Dashboards and in reports.

## The Test Agent

The Test Agent is the communication tool between QAComplete's Test Management modules and your automated testing tools. Currently, the following testing tools are supported:

- TestComplete (TestExecute) by SmartBear.
- Ready! API by SmartBear.
- SoapUI by SmartBear.
- Selenium WebDriver for the following unit testing frameworks:
  - JUnit
  - TestNG
  - NUnit

The Test Agent needs to be installed and configured on each computer that will run automated tests so that results are posted to Test Management. The Agent is also used for scheduled automation runs.

When you install the Test Agent on a Host machine, the Agent will automatically create a Test Host record in Test Management.

**Note:** You can also use Selenium Test Agents to automate running of ordinary unit tests (that do not use Selenium modules) created with JUnit, TestNG and NUnit unit testing frameworks.

### Requirements

To install the Test Agent, your test machine must meet the following requirements:

- Have an active internet connection.
- To run TestComplete automated tests:
  - TestComplete or TestExecute version 9.0 or later installed.
- To run automated API tests:
  - Ready! API version 1.2.0 installed.
- -- or --
- SoapUI version 5.1.2 installed.
- To run tests with a Java unit testing framework:
  - Java Runtime Environment (JRE) 5 - 8 installed.
  - Appropriate Java unit testing framework, **JUnit** or **TestNG**, installed and configured.

- To run Selenium automated web tests created with an appropriate Java unit framework:
  - Selenium WebDriver modules installed and configured in your JUnit or TestNG framework.
- If you want to use Maven to run tests:
  - Maven 3.2.3 or later modules installed and configured.
  - The PATH environment variable on the test machine must specify the path to the Maven modules.
- To run tests with NUnit testing framework:
  - NUnit unit testing framework installed.
- To run Selenium automated web tests created with NUnit:
  - Selenium WebDriver modules installed and configured in NUnit framework.

### Active Directory Authentication Restriction (On-Premises Only)

To register the Test Agent on a computer, do not use an account with Active Directory Authentication. Use a non-Active Directory user account.

You can enable and disable Active Directory Authentication for a user in **Setup > Security > Users** in QAComplete.

### Default Ports

By default, the ports used by the Test Agent and related processes are:

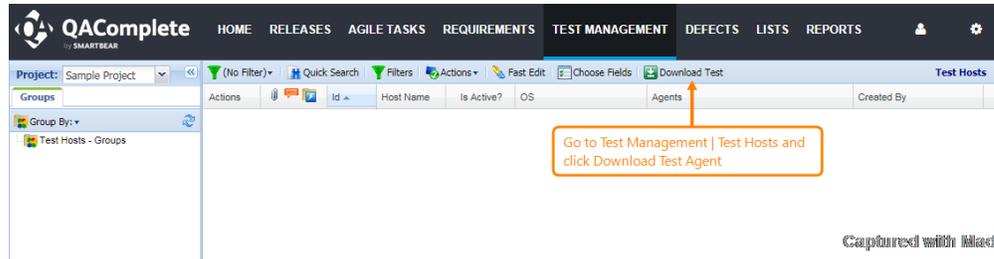
- TestComplete (and TestExecute) license manager: Port 1947.
- For TestComplete and TestExecute Network Suite, the port depends on the product version:
  - In TestComplete (and TestExecute) version 11: ports 6090-6092.
  - In versions 10.20 - 10.60: ports 6020-6022.
  - In versions 9.10 - 10.10: ports 6090-6092.

To find the ports used by Network Suite in TestComplete (or TestExecute), see the documentation for your version of TestComplete (or TestExecute).

- Test Agent: Ports 80 and 443.

### Installing the Test Agent

- Download the Agent installation file:
  - Log in to QAComplete.
  - Go to the **Test Management > Test Hosts** page.
  - Click **Download Test Agent**.



- Run the installation and follow the Installation Wizard instructions.
- On the **Register Test Agent in QAComplete** page of the wizard, register the agent in QAComplete.
- In the **Web** text box, enter the address of your QAComplete web service.
- For SaaS clients, the address is `https://rest.qacomplete.smartbear.com/rest-api/service`
- For On-Premise clients, the address is: `http://[your server name]/rest-api/service` (where [your server name] is your QAComplete server).
- In the **User** and **Password** text boxes, enter the email address and the password that will be used to log in to QAComplete account.  
**Note For On-Premises Only:** You cannot register the Test Agent with an account that uses Active Directory Authentication. Register the Test Agent using a non-Active Directory account.
- If a proxy server is used in your network, configure Test Agent to connect to the network through that proxy server:
  - Click **Proxy Settings**.
  - In the **Proxy Settings** dialog, enter the network name or IP address, and the port of the proxy server that Test Agent will use to connect to QAComplete. For example, `192.168.100.50:8888`.
  - If your proxy server requires user authentication, enter the name and password of the user account under which you will connect to the proxy server.
  - Click **OK** to close the dialog and continue installing Test Agent.
- Click **Next** to continue.
- If the specified QAComplete account belongs to multiple departments (that is, this user sees a list of departments when logging in), the **Select Department** page appears. Select the department whose tests you want to run on this computer and click **Next**.
- On the **Logon** page of the installation wizard, enter the credentials of the user that will run your automated tests on this machine.
  - In the **User Name** and **Password** text boxes, enter the name and the password of the user account.
- Click **Next** to complete the installation.

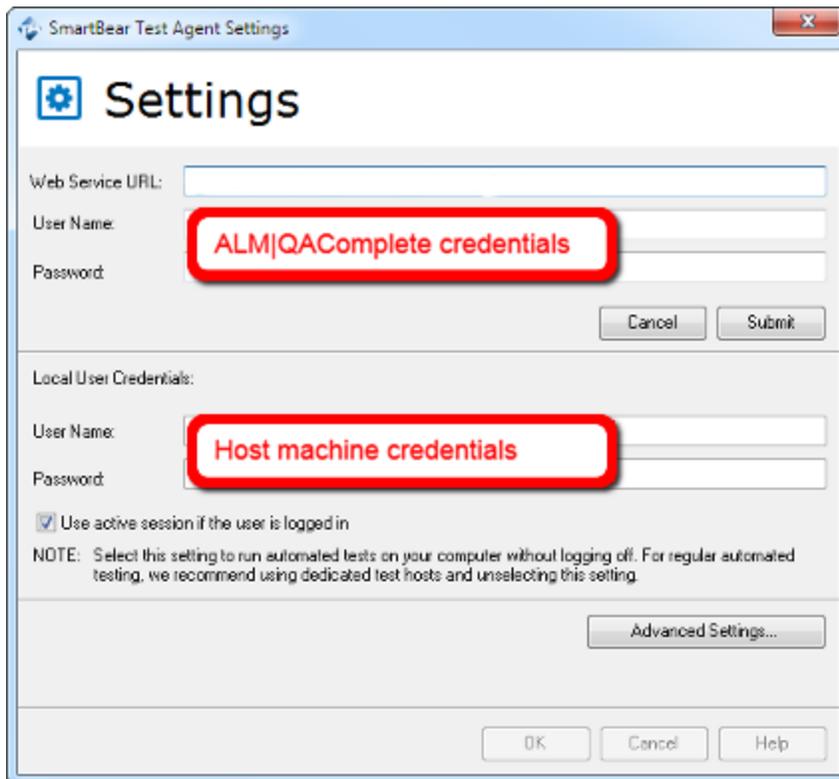
After you install Test Agent on your test computer, the Agent will automatically add information about the host computer to the **Test Hosts** screen in QAComplete's Test Management.

During the installation, Test Agent checks whether **TestComplete** or **TestExecute** is installed on the Test Host computer. If any of them is, that information will appear in Agents field. **JUnit (Selenium)**, **NUnit (Selenium)**, **TestNG (Selenium)**, **Ready! API / SoapUI** tools are always considered as installed, so Test Agent automatically adds information about them to the Agents field of the corresponding Test Host without any checking.

You cannot run Selenium or Unit tests without initial configuration. For information on configuration process, see [Configuring the Test Agent](#).

## Configuring the Test Agent

Once the Test Agent is installed on your test computer, you can check or modify the Agent configuration.



On your Test Host machine:

- Go to Start - All Programs - SmartBear - Test Agent - Test Agent Settings.

### Configuring QAComplete Web Service Address

- In the **Web Service URL** field, enter the Address of the REST Service:
  - For Software as a Service customers, the address is
    - <https://rest.qacomplete.smartbear.com/rest-api/service>
  - For On-Premises customers, the address must point to the Agent on your server
    - [http://\[your server name\]/rest-api/service](http://[your server name]/rest-api/service)
- To change the QAComplete login credentials, click **Change** and enter the email address and password to use to log in to your QAComplete account. Note: for security purposes, when you click Change, you must enter new credentials.
- Click **Submit**.  
Note: If the specified QAComplete account belongs to multiple departments, you will be prompted to select a department to use this computer with.

### Configuring Test Host Credentials

- To change login credentials to the Host machine, under **Local User Credentials**, enter the name and password for the Windows user account that will run the automated test from this machine. If the account belongs to a domain, make sure to enter the domain name along with the user name ([my domain/my domain user name]).
- Set the **Use active session** option by checking or unchecking the box.
  - If checked: the Test Agent will check if the Local User is currently logged in to the Host machine. If so, it will not perform the logout process. If this user is not logged in, the Agent will close any open applications, log out of the Host machine, log in as the user specified here, run the test, and log out.
  - If not checked: the Test Agent will always close open applications on the Host machine, log out, log in, run the test, and log out.
- Click **OK** to save your changes.

### Configuring Test Agent to Run Selenium and Unit Tests with JUnit and TestNG

- To set paths needed to run tests with Java unit testing frameworks, click **Advanced Settings**.
- In the **Advanced Settings** dialog, open the **Test Runners** page.
- Enter the path to the Java Runtime executable (*java.exe*).  
For example, if you have several Java versions installed, specify the path to the Java Runtime version you will use to run tests.
- In the **Java Class Path** field, enter the path to the required framework library (junit-n.nn.jar or testng-n.n.n.jar). (You can skip this step if you are going to use Maven to run tests).
- Click **OK** to save your changes.

#### Configuring Test Agent to Run Selenium and Unit Tests with NUnit

- To set the path needed to run tests with NUnit testing framework, click **Advanced Settings**.
- In the **Advanced Settings** dialog, open the **Test Runners** page.
- Enter the path to the NUnit executable (*nunit-console.exe*).
- Click **OK** to save the changes.

#### Configuring Test Agent to Run Ready! API/SoapUI Tests

- To set the path to the Ready! API or SoapUI test runner tool, click **Advanced Settings**.
- In the **Advanced Settings** dialog, open the **Test Runners** page.
- Enter the path to the test runner tool (*testrunner.bat*) that is installed with Ready!API or SoapUI. By default, Test Agent will search for the test runner tool in the folder where Ready! API/SoapUI is installed.
- Click **OK** to save the changes.

#### Configuring Test Agent to Use Proxy

- To configure Test Agent to communicate with QAComplete Test Management modules through a proxy server, click **Advanced Settings**.
- In the **Advanced Settings** dialog, open the **Proxy Settings** page.
- Select the **Use Proxy for connection** check box.
- In **Proxy Address**, enter the network name or IP address, and port of your proxy server. For example, *192.168.100.50:8888*.
- If your proxy server requires user authentication, enter the user name and the password of the user account under which you will connect to the proxy server.
- Click **OK** to save the changes.

#### Configuring Test Agent to Send Usage Information

If you want to help us to improve Test Agent, please allow it to collect information on how you use Test Agent and to send that information to SmartBear.

- In **Test Agent Settings**, click **Advanced Settings**.
- In the **Advanced Settings** dialog, open the **Misc** page.
- Select the **Send usage data to help improve TestAgent** check box.
- Click **OK** to save the changes.

If you permit data collection, Test Agent will collection information on how you use the product and will send it to our server every time Test Agent closes or when an error occurs. SmartBear will not share the collected information with any third-parties.

Test Agent will not collect any personal information, like password, logins and other sensitive information.

Information on how you work with our product will help us to understand how we can improve it.

## About Test Hosts

A Test Host is a computer that runs your automated tests. Each Test Host must have a supported automated testing tool and the Test Agent installed and configured.

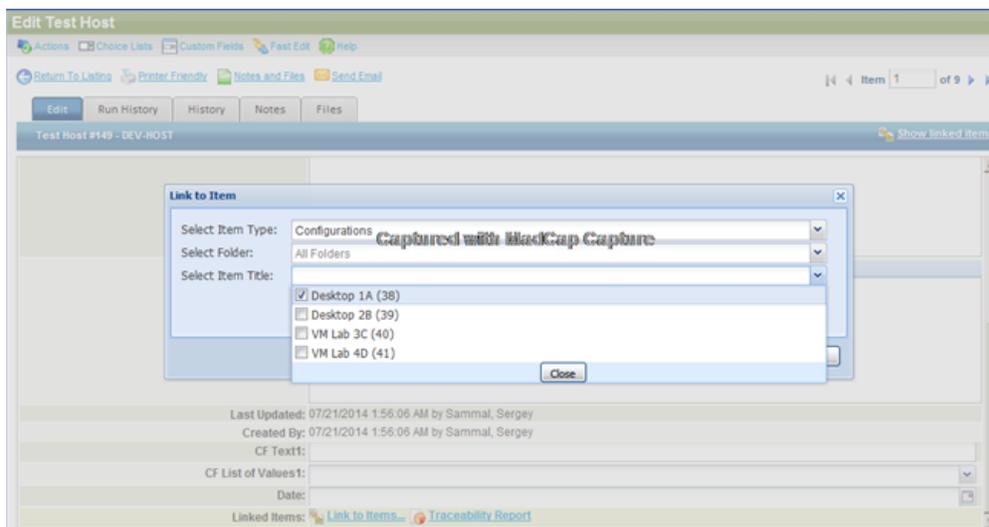
Once the Test Agent is installed and configured on your Test Host, information about the computer is automatically relayed to Test Management as a new Test Host. You can view the list of Test Hosts from Test Management / Test Hosts.

Other than the Test Host Description and Notes, you cannot add nor edit Test Hosts from within Test Management. The Test Agent provides all information about the Test Host directly to QAComplete when the Test Agent is configured on the machine.

Note: it is important to ensure that any dependent libraries, data for data driven tests, etc. exist on the appropriate Hosts so that your automated test can locate all necessary pieces to run successfully.

To include automated test results on Test Management reports that are filtered by Configurations, link your Hosts to the appropriate configuration.

- Open your Test Host record
- Click Show Linked Items or scroll to the bottom of the form
- Click Link to Items
- Configurations should be selected in the Select Item Type
- Click the drop down in Select Item Title and check the Configurations that you want associated with the Test Host.
- Click Close, then click Submit.



- Click Submit to close the edit Test Host form.

Note: Only one Configuration will be used for reporting, so match one Test Host to one Configuration.

## Configuring Virtual Machine Hosts

If you run your TestComplete automated tests on a virtual machine, we recommend that you disable automatic log opening after test runs to improve performance.

- If you have TestComplete, select Tools | Options from the main menu.
- If you have TestExecute, right-click the TestExecute icon in the notification area and select Options.
- In the dialog that opens, select Engines | Log.
- Clear the *Activate after test run* check box and click OK.

## Creating Automated Tests

In the Test Library, you create Automated Tests by assigning automated or unit tests created with one of the [supported testing tools or frameworks](#) to your Tests.

Learn how to assign the following test types to your Test Library Tests:

- [TestComplete Tests](#)
- [Ready! API and SoapUI Tests](#)
- [JUnit Tests and Selenium JUnit Tests](#)
- [TestNG Tests and Selenium TestNG Tests](#)
- [JUnit Tests and Selenium NUnit Tests](#)

When you associate an automated (or unit) test with the Test, Test Management automatically recognizes that the Test is Automated and selects the Automated check box. You can use the choice list for Execution Type to also designate a test as Automated, or modify the Execution Type choice list values for anything that is meaningful to you.

**Note:** When running Automated Tests, any Steps that may be on the Test are ignored.

To run your Automated Test on demand, please see [Running Automated Tests](#).

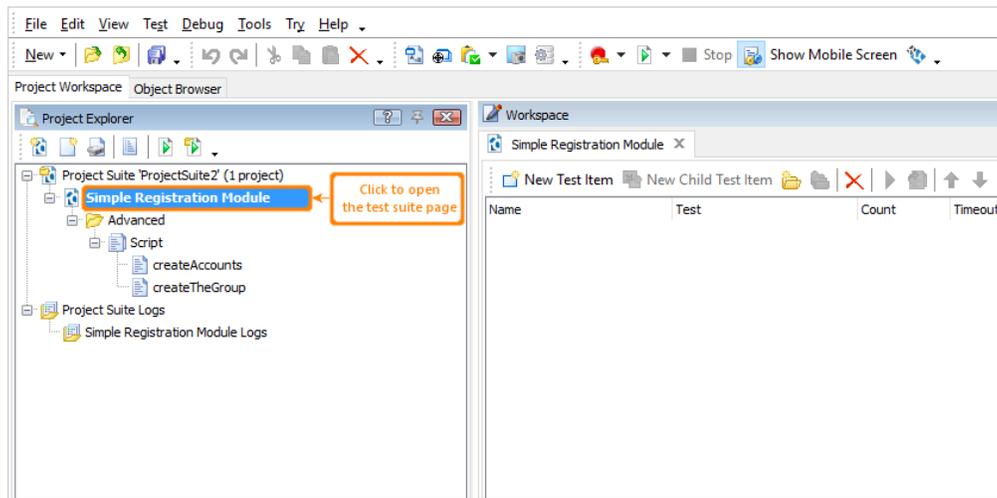
To Schedule your Automated Test to run automatically, please see [Scheduling Automated Tests](#).

## Assigning TestComplete Tests to Tests

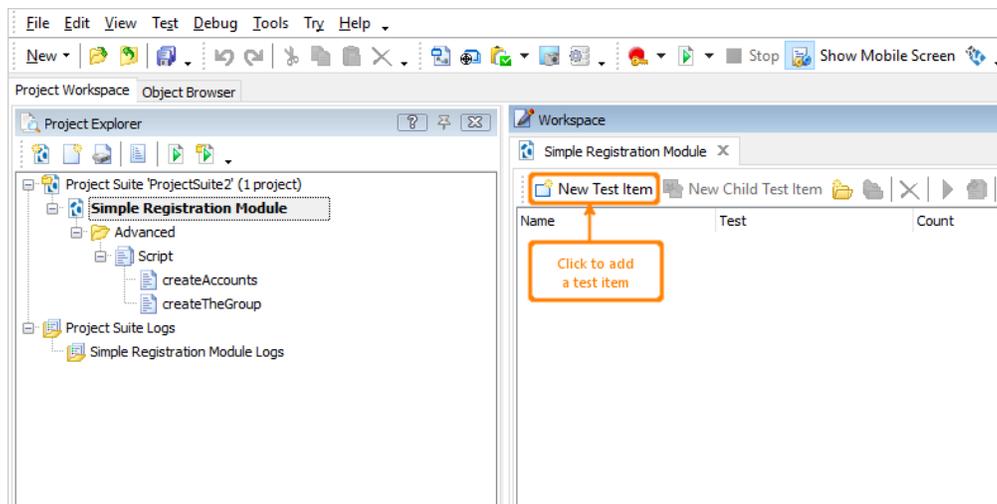
### Preparing TestComplete Project Suite

1. Create tests in your TestComplete project.
2. **Create and organize test items that will run those tests. To do so:**

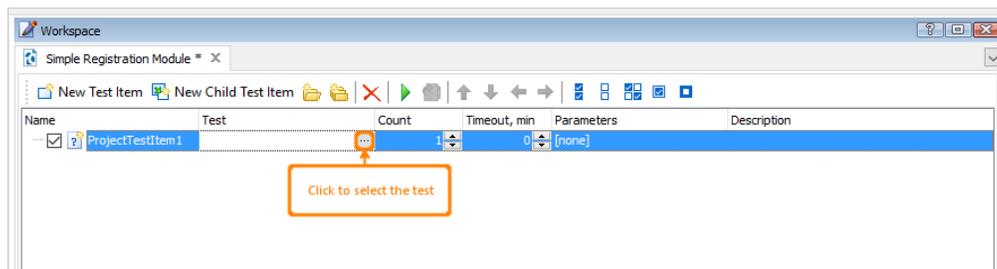
1. Click the test suite in the **Project Explorer** to open its page.



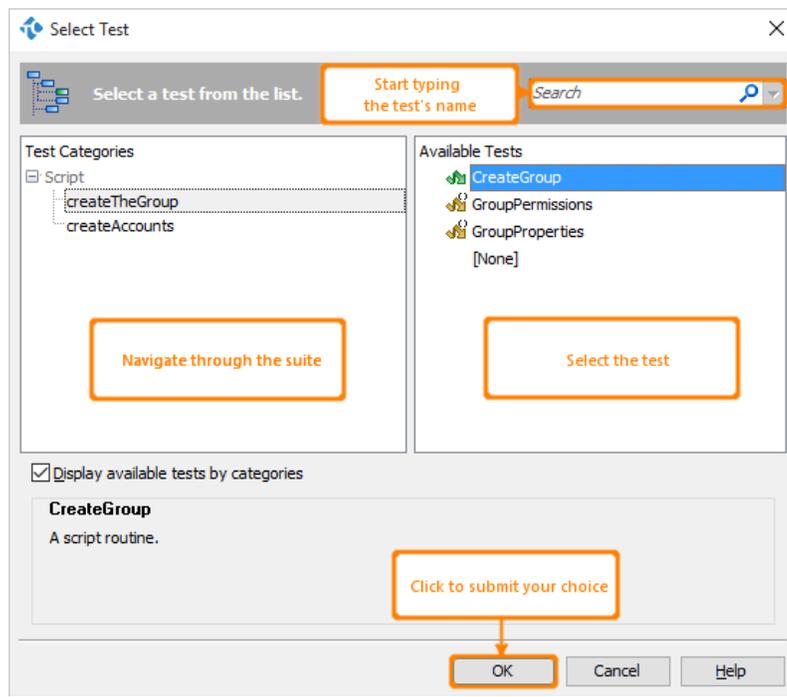
2. Click **New Test Item** to add the first test item.



3. Click ... in the **Test** column to open the **Select Test** dialog.



4. Select the needed test in the dialog. Use the **Test Categories** and **Available Tests** fields, or search for the test with the **Search** field. Click **OK** to submit your selection.



5. Add more test items and child test items if needed and arrange them as you like.
  6. Save the changes you have applied.
3. If needed, zip up your TestComplete project suite. Include all files and folders that belong to the project suite.  
**Note:** Your TestComplete project must not reference files outside the project folder.
  4. Make note of where the project suite or the archive (zip) file resides. You can place the project suite or the archive on a local folder on the test host where it will run or in a shared network folder. Make sure that the project suite can access all need test files. Alternatively, you can upload the archive to the server (see below).

### Assigning TestComplete Projects

You can assign TestComplete tests to your QAComplete tests on the **Automation** page or, if you use TestComplete version 12, you can add your TestComplete tests to QAComplete directly from TestComplete IDE.

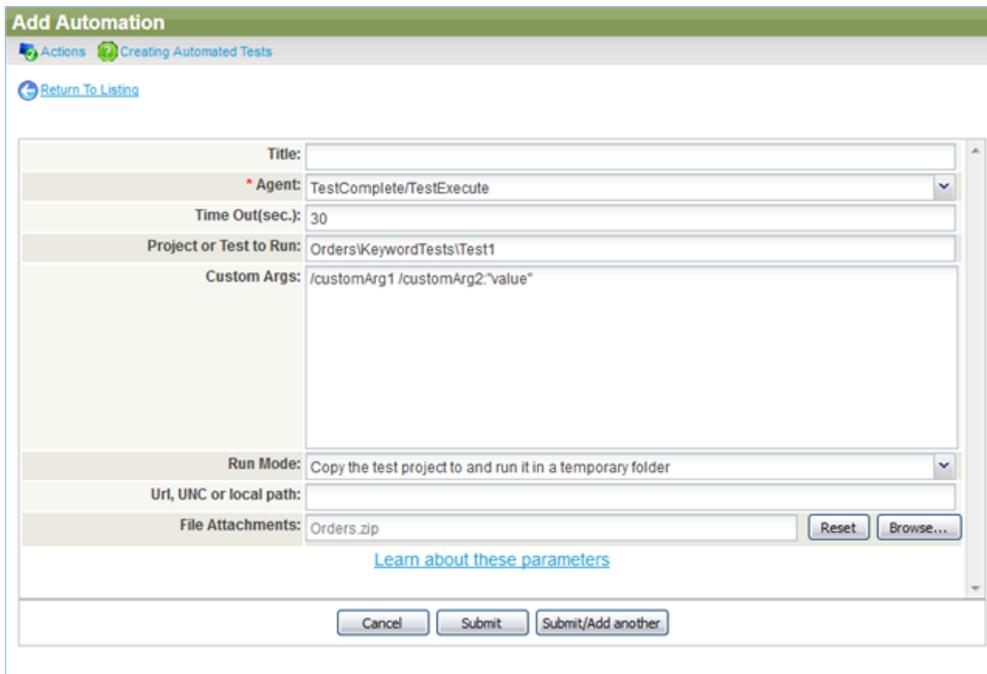
## Adding TestComplete Tests Manually

### 1. Preparing Your Test Library Test

- In **Test Library**, open an existing Test for the TestComplete project suite or add a new Test.
- Select a **Default Host Name**. If you skip this step, after you assign an automated test and click Submit, the system will remind you that the Test Host is missing.
- Save the changes.

### 2. Assigning TestComplete Test to the Test

- On the **Automations** tab, click **Add New** to add an automated Test.



The screenshot shows the 'Add Automation' dialog box with the following fields and values:

- Title:** (Empty)
- Agent:** TestComplete/TestExecute
- Time Out(sec.):** 30
- Project or Test to Run:** Orders\KeywordTests\Test1
- Custom Args:** /customArg1 /customArg2:"value"
- Run Mode:** Copy the test project to and run it in a temporary folder
- Uri, UNC or local path:** (Empty)
- File Attachments:** Orders.zip

Buttons at the bottom include: Cancel, Submit, Submit/Add another, Reset, and Browse...

- In the **Agent** drop-down list, select **TestComplete/TestExecute**.
- You can leave the **Title** blank, and the system will use the name of the specified TestComplete file, or you can enter a meaningful Title.
- Enter a time out value in seconds. If the test takes longer than the given time out, the test runner will stop running.
  - The Test Runner passes the timeout value to TestComplete/TestExecute as a run parameter. The minimum value is 30 seconds: if you specify a lower value, it still uses 30 seconds. If the test takes longer than the specified time out, it will log an error and end. For more information about how TestComplete uses the time out, please see [TestComplete Command Line](#).

In **Url, UNC or local path**, enter the path to your TestComplete project suite or to the archive (zip) that contains the project suite.

The path format depends on the project suite location. Below are examples of possible values:

Value	Description
C:\MyTests\MyProjectSuite.pjs -- or -- file:///C:\MyTests\MyProjectSuite.pjs	A project suite (.pjs) residing in the local folder on the test host.
C:\MyTests\ -- or -- file:///C:\MyTests\	A local folder on the test host where a project suite resides. Note: if there is more than one project suite in the folder, an error will occur.
C:\MyTests\MyProjectSuite.zip -- or -- file:///C:\MyTests\MyProjectSuite.zip	An archive (zip) file that contains the project suite and resides in a local folder on the test host.
\\Tests-PC\SharedFolder\MyProjectSuite.pjs -- or -- file:///Tests-PC\SharedFolder\MyProjectSuite.pjs	A project suite (.pjs) residing in a shared folder on a computer in your network.
\\Tests-PC\SharedFolder\ -- or -- file:///Tests-PC\SharedFolder\	A shared network folder where the project suite resides. Note: if there is more than one project suite in the folder, an error will occur.
\\Tests-PC\SharedFolder\MyProjectSuite.zip -- or -- file:///Tests-PC\SharedFolder\MyProjectSuite.zip	An archive (zip) file that contains the project suite and resides in a shared network folder.
http://downloads/mytests.com/MyProjectSuite.zip	A web site URL path to the archive (zip) file that contains the project suite.

**Note:** If the path you entered is incorrectly formatted, QAComplete will highlight the path in red.

Or, simply use **File Attachments** to upload the archive (zip) file that contains the project suite. Click **Browse** and navigate to the zipped file.

- The **Run Mode** specifies how the test agent will get your TestComplete test files and run the test.
  - To copy test files to the test host and run the test in a temporary folder there, choose **Copy the test project to and run it in a temporary folder** in the Run Mode drop-down list.
 

If the **Url, UNC or local path** specifies the UNC path to a project suite file (.pjs), all the contents of the folder where the project suite file resides will be copied to the test host.
  - To run the test directly in the folder where the test files reside, choose **Run the test project in place (if possible)** from the Run Mode drop-down list. If the test files reside in a shared network folder or in a local folder on the test host, the test agent will run the test directly in that folder.
- The **Project or Test to Run** field is used if you want to run a portion of your TestComplete project suite:
  - To run the entire project suite, leave the **Project or Test to Run** field blank.
  - To run a specific project within the project suite, enter the project name.
  - To run a specific keyword test, script or project item, enter the path to the item.

Below are examples of possible field values:

Value	Description
<i>(no value)</i>	All test items in all projects in the project suite.
<i>TestProject1</i>	All test items in the TestProject1 project.
<i>TestProject1\KeywordTests\Test1</i>	The Test1 keyword test in the TestProject1 project.
<i>TestProject1\Script\Unit1\Test1</i>	The Test1 script routine in the Unit1 script unit of the MyProject project.

- In the **Custom Args** field, you can enter custom command line parameters to pass to TestComplete (TestExecute).
- Click **Submit**.

### Exporting TestComplete Tests From TestComplete IDE

#### Requirements

- You must have TestComplete **12** installed.
- The **QAComplete Integration** plugin must be installed and enabled in TestComplete. (The plugin is installed and enabled **automatically** when you install TestComplete.)

## 1. Uploading Test From TestComplete

- On your computer, start TestComplete and open a project suite that contains tests you want to add to QAComplete.
- In **Project Explorer** in TestComplete, right-click the project suite, or a project, or a test that you want to export and then click **Export to QAComplete**.

TestComplete will open the **Export TestComplete Test to QAComplete** wizard. Follow the wizard instructions to add your tests to QAComplete.

- Enter the address of your QAComplete server.  
For SaaS clients, the address is `https://login.qacomplete.smartbear.com/`.  
For On-Premise clients, the address is: `https://[your server name]/` (where [your server name] is your QAComplete server).  
Enter the email address and the password that will be used to log in to the QAComplete account.
- Select a QAComplete project to which you want to add tests.
- Enter the test's title and description. If needed, select a folder in your Test Library, to which you want to add a test.

Specify the test's timeout.

- The wizard will create a new test in QAComplete. Specify test parameters, if needed.

You also can find information on exporting TestComplete tests to QAComplete in the TestComplete Documentation.

The TestComplete test will be added to your QAComplete project's Test Library, to the folder you selected during the exporting.

To view and configure the added test, in QAComplete, open the added test and switch to the **Automations** page.

## Assigning Ready! API and SoapUI Tests to Tests

### 1. (Optional) Configure the Test Agent

On the Test Host that will run the test:

- Launch the Test Agent Settings tool (Start | All Programs | SmartBear | TestAgent | Test Agent Settings).
- Click **Advanced Settings**.

- If needed, specify the path to the test runner tool (*testrunner.bat*) that will run the API tests. The tool is installed with Ready! API or SoapUI and resides in the folder where Ready! API or SoapUI is installed.

## 2. Prepare Your API Test

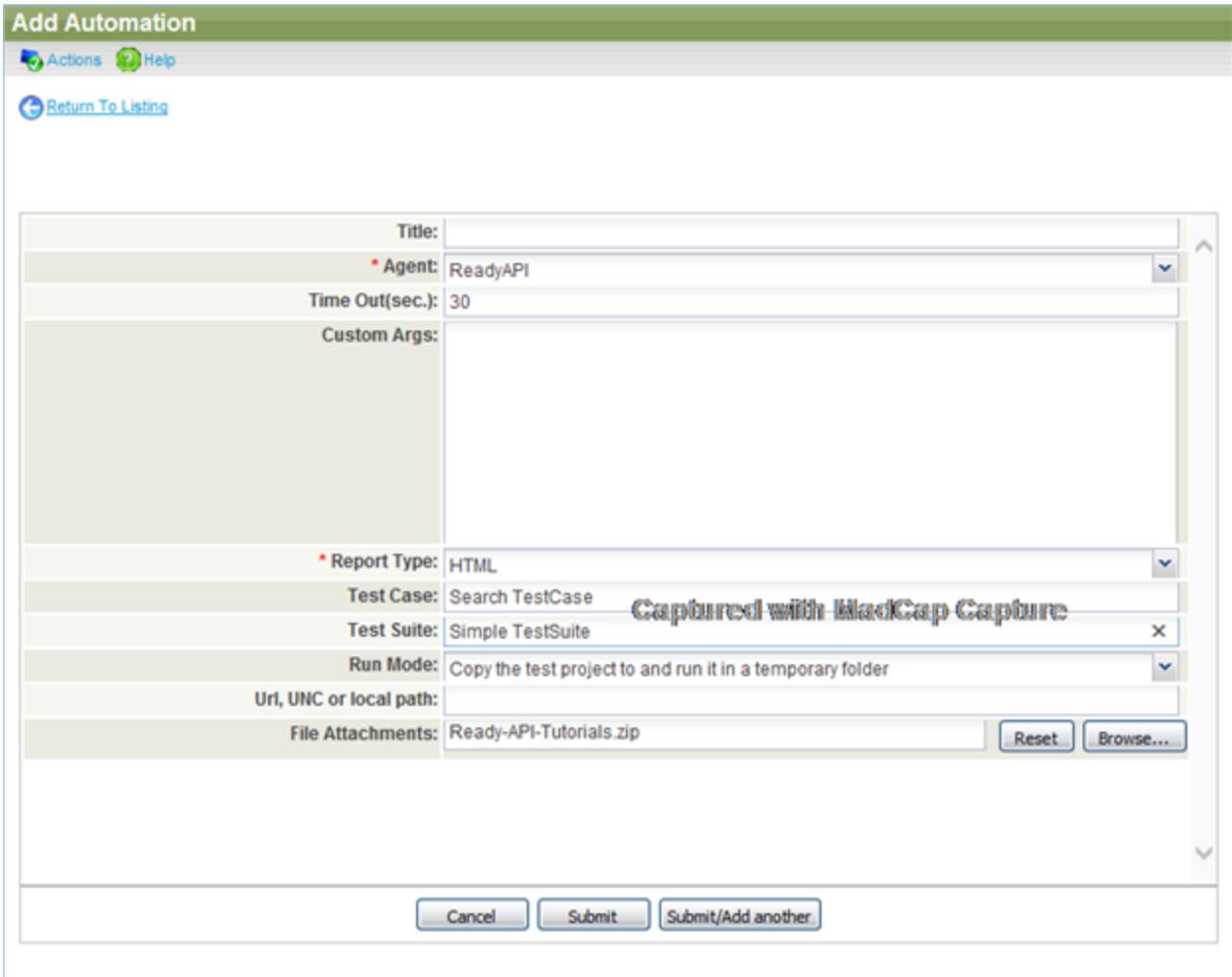
- If needed, zip up your API test project.
- Make note of where the project or the zipped project resides. You can place the project or zipped project in a local folder on the test host machine that will run the project or in a shared network folder. Alternatively, you can upload the project or an archive to the server (see below).

## 3. Prepare Your Test Library Test

- In **Test Library**, open an existing Test for the Ready! API or SoapUI test or add a new Test.
- Select a **Default Host Name**. If you skip this step, after you assign an automated test and click Submit, the system will remind you that the Test Host is missing.
- Save the changes.

#### 4. Assign the API Test to the Test

- On the **Automations** tab, click **Add New** to add an automated test.



The screenshot shows the 'Add Automation' form with the following fields and values:

- Title:** (empty)
- Agent:** ReadyAPI
- Time Out(sec.):** 30
- Custom Args:** (empty text area)
- Report Type:** HTML
- Test Case:** Search TestCase
- Test Suite:** Simple TestSuite
- Run Mode:** Copy the test project to and run it in a temporary folder
- Url, UNC or local path:** (empty)
- File Attachments:** Ready-API-Tutorials.zip

Buttons at the bottom: Cancel, Submit, Submit/Add another.

- In the **Agent** drop-down list, select **Ready! API / SoapUI**.  
Note: If the Default Host assigned to the Test does not support the selected automated testing tool, the test will not run.
- You can leave the **Title** blank, and the system will use the name of the attached test, or you can enter a meaningful Title.
- Enter a time out value in seconds. If the test takes longer than the given time out, the test runner will stop running.
- To run a specific test suite in your project, enter its name in **Test Suite**. For example, *Simple TestSuite*.

- To run a specific test case in your project, enter its name in **Test Case**. For example, *Search TestCase*.
- In **Report Type**, choose the format for exporting the test results report. If you use SoapUI Free to run API tests, choose **No Report**.
- In **Custom Args**, enter additional command line arguments to pass to the *testrunner* tool that will run your test. You can find information on command line arguments in the Ready! API or SoapUI documentation.
- In **Url, UNC or local path**, enter the path to your Ready! API or SoapUI project (.xml) or to the archive (.zip) that contains your project.  
Below are examples of possible paths:

Value	Description
C:\MyTests\MySOAPPProject.xml -- or -- file:///C:\MyTests\MySOAPPProject.xml	A project (.xml) residing in the local folder on the test host.
C:\MyTests\ -- or -- file:///C:\MyTests\	A local folder on the test host where the project to run resides. Note: If there is more than one project in the specified folder, an error will occur.
C:\My Tests\MySOAPPProject.zip -- or -- file:///C:\My Tests\MySOAPPProject.zip	The archive (zip) file that contains the project, and resides in a local folder on the test host.
\\Tests-PC\SharedFolder\MySOAPPProject.xml -- or -- file:///Tests-PC\SharedFolder\MySOAPPProject.xml	A project (.xml) residing in a shared folder on a computer in your network.
\\Tests-PC\SharedFolder\ -- or -- file:///Tests-PC\SharedFolder\	A shared network folder where the project resides. Note: If there is more than one project in the folder, an error will occur.
\\Tests-PC\SharedFolder\MySOAPPProject.zip -- or -- file:///Tests-PC\SharedFolder\MySOAPPProject.zip	The archive (zip) file that contains the project, and resides in a shared network folder.

Value	Description
<code>http://downloads/mytests.com/MySOAPProject.zip</code>	A web site URL path to the archive (zip) file that contains the project.

**Note:** If path you enter is incorrectly formatted, QAComplete will highlight the path in red.

- Or, simply upload your project or zipped file using **File Attachments**. Click **Browse** and navigate to the desired file.
- The **Run Mode** specifies how the test agent will get your test files and run the test.
  - To copy test files to the test host and run the test in a temporary folder there, choose **Copy the test project to and run it in a temporary folder** in the Run Mode drop-down list.
  - To run the test directly in the folder where the test files reside, choose **Run the test project in place (if possible)** from the Run Mode drop-down list. If the test files reside in a shared network folder or in a local folder on the test host, the test agent will run the test directly in that folder.
- Click **Submit**.

## Assigning Selenium JUnit and JUnit Unit Tests to Tests

**Important:** You can use JUnit (Selenium) Test Agent to automate running of both Selenium JUnit tests and ordinary JUnit unit tests that do not use Selenium modules. The procedure of assigning tests to Test Library Tests is the same for both test types.

This instruction demonstrates how to assign a Selenium JUnit tests to a Test Library Test. Use the same procedure to assign an ordinary JUnit unit test to a Test.

### 1. Configure the Test Agent

On the Test Host that will run the test:

- Launch the Test Agent Settings tool (Start | All Programs | SmartBear | TestAgent | Test Agent Settings).
- Click **Advanced Settings**.
- If needed, specify the path to the Java Runtime executable (*java.exe*). For example, if you have several Java versions installed, specify the path to the Java Runtime version you will use to run tests.
- Specify the path to the *junit-n.nn.jar* file location (you can skip this step if you use Maven to run tests).

## 2. Prepare your Selenium WebDriver Test

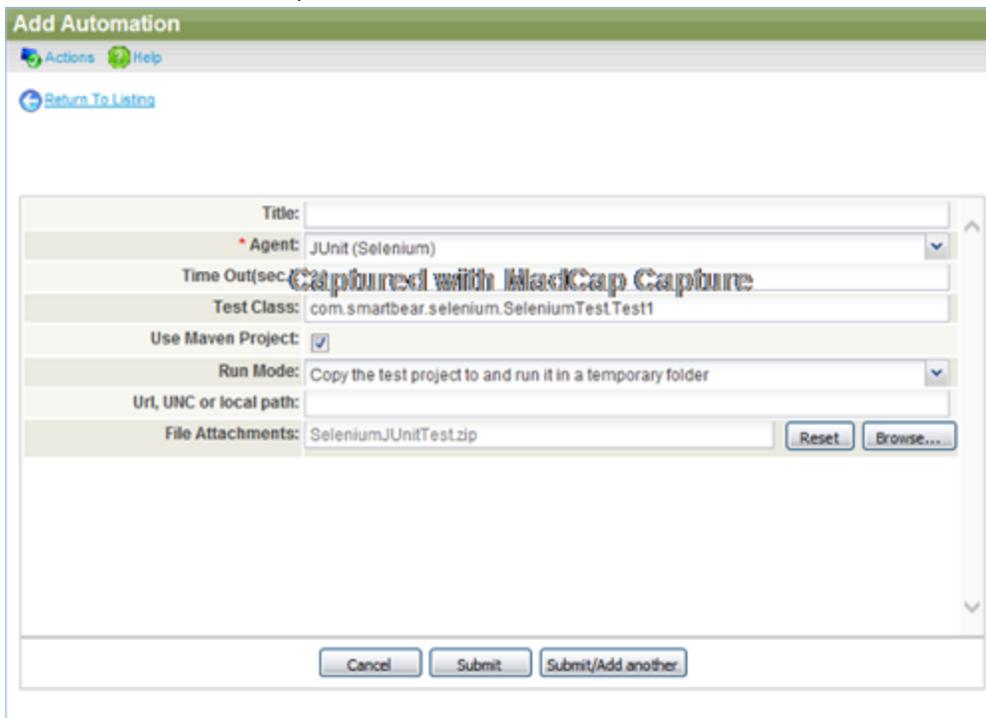
- If needed, zip up the folder where your Selenium WebDriver tests created with JUnit reside.
- Make note of where the test files or archive (zip) file reside.  
You can place the test files (or the zip file that contains the test files) in a local folder on the test host where the tests will run or in a shared network folder.  
Alternatively, you can upload the archive file to the server (see below).

## 3. Prepare Your Test Library Test

- In **Test Library**, open an existing Test for the Selenium WebDriver test or add a new Test.
- Select a **Default Host Name**. If you skip this step, after you assign an automated test and click **Submit**, the system will remind you that the Test Host is missing.
- Save the changes.

## 4. Assign Selenium WebDriver Test to the Test

- On the **Automations** tab, click **Add New** to add an automated test.



- In the **Agent** drop-down list, select **JUnit (Selenium)**.  
Note: If the Default Host assigned to the Test does not support the selected automated testing tool, the test will not run.

- You can leave the **Title** blank, and the system will use the name of the attached Selenium test, or you can enter a meaningful Title.
- The **Timeout** tells the Test Runner how long to let the test run. Enter the time out value in seconds. If the test takes longer than the given time out, the test runner will stop running.
- In **Url, UNC or local path**, enter the path to the folder where your Selenium WebDriver test files reside or to the archive (.zip) that contains the test files.

Below are examples of possible paths:

Value	Description
C:\MyJUnitTests\ -- or -- file:///C:\MyJUnitTests\	A local folder on the test host where tests reside.
C:\MyJUnitTests\MyJUnitTest.zip -- or -- file:///C:\MyJUnitTests\MyJUnitTest.zip	The archive (zip) file that contains the test files and resides in a local folder on the test host.
\\Tests-PC\SharedFolder\ -- or -- file:///Tests-PC\SharedFolder\	A shared network folder where tests reside.
\\Tests-PC\SharedFolder\MyJUnitTest.zip -- or -- file:///Tests-PC\SharedFolder\MyJUnitTest.zip	The archive (zip) file that contains the tests and resides in a shared network folder.
http://downloads/mytests.com/MyJUnitTest.zip	A web site URL path to the archive (zip) file that contains the test files.

**Note:** If path you enter is incorrectly formatted, QAComplete will highlight the path in red.

- Alternately, upload your zipped file using **File Attachments**. Click **Browse** and navigate to the zipped test files.
- The **Run Mode** specifies how the test agent will get your test files and run the test.
  - To copy test files to the test host and run the test in a temporary folder there, choose **Copy the test project to and run it in a temporary folder** in the Run Mode drop-down list.
  - To run the test directly in the folder where the test files reside, choose **Run the test project in place (if possible)** from the Run Mode drop-down list. If the test files reside in a shared network folder or in a local folder on the test host, the test agent will run the test directly in that folder.
- If you want to run your Selenium test by using Maven, select **Use Maven Project**.

- Enter the test case class name in **Test Class** field.  
If you use Maven to run tests, you can leave the field empty to run all test cases defined in your Selenium test.
- Click **Submit**.

## Assigning Selenium TestNG Tests to Tests

**Important:** You can use TestNG (Selenium) Test Agent to automate running of both Selenium TestNG tests and ordinary TestNG unit tests that do not use Selenium modules. The procedure of assigning tests to Test Library Tests is the same for both test types.

This instruction demonstrates how to assign a Selenium TestNG tests to a Test Library Test. Use the same procedure to assign an ordinary TestNG unit test to a Test.

### 1. Configure the Test Agent

On the Test Host that will run the test:

- Launch the Test Agent Settings tool (Start | All Programs | SmartBear | TestAgent | Test Agent Settings).
- Click **Advanced Settings**.
- If needed, specify the path to the Java Runtime executable (*java.exe*). For example, if you have several Java versions installed, specify the path to the Java Runtime version you will use to run tests.
- Specify the path to the *testng-n.n.n.jar* file location (you can skip this step if you will use Maven to run tests).

### 2. Prepare your Selenium WebDriver Test

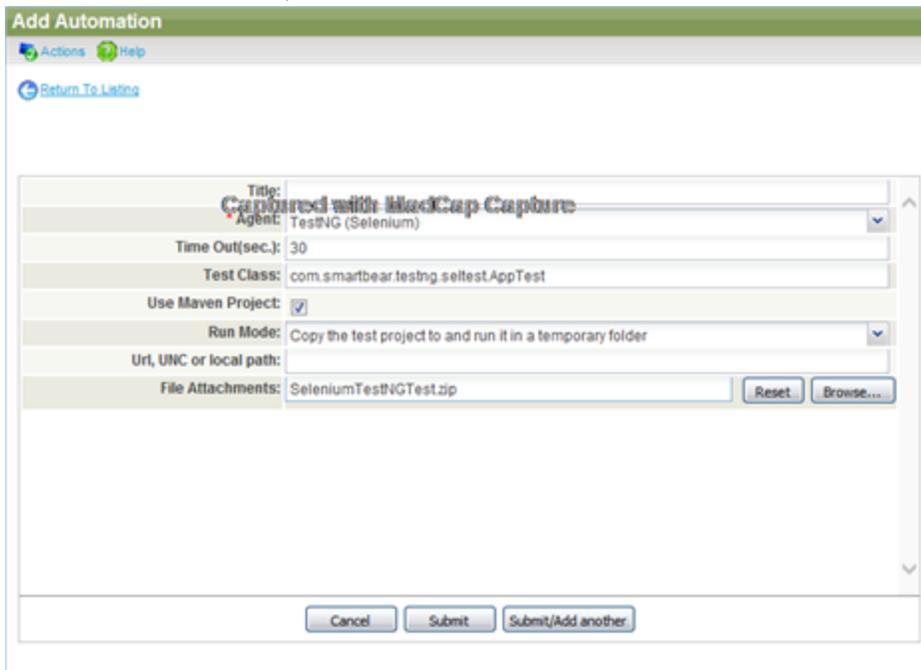
- If needed, zip up the folder where your Selenium WebDriver tests created with TestNG reside.
- Make note of where the test files or the archive (zip) file reside.  
You can place the test files or the archive file on a local folder on the test host where the tests will run or in a shared network folder.  
Alternatively, you can upload the archive to the server (see below).

### 3. Prepare Your Test Library Test

- In **Test Library**, open an existing Test for the Selenium WebDriver test or add a new Test.
- Select a **Default Host Name**. If you skip this step, after you assign an automated test and click **Submit**, the system will remind you that the Test Host is missing.
- Save the changes.

#### 4. Assign Selenium WebDriver Test to the Test

- On the **Automation** tab, click **Add New** to add an automated test.



- In the **Agent** drop-down list, select **TestNG (Selenium)**.  
Note: If the Default Host assigned to the Test does not support the selected automated testing tool, the test will not run.
- You can leave the **Title** blank, and the system will use the name of the attached Selenium test, or you can enter a meaningful Title.
- The **Timeout** tells the Test Runner how long to let the test run. Enter the time out value in seconds. If the test takes longer than the given time out, the test runner will stop running.
- In **Url, UNC or local path**, enter the path to the folder where your TestNG test files reside or to the archive (zip) file that contains the test files.  
Below are examples of possible path values:

Value	Description
C:\MyTestNGTests\ -- or -- file:///C:\MyTestNGTests\	A local folder on the test host where tests reside.
C:\MyJUnitTests\MyTestNGTest.zip	The archive (zip) file that contains

Value	Description
-- or -- <i>file:///C:\MyJUnitTests\MyTestNGTest.zip</i>	the test files and resides in a local folder on the test host.
\\Tests-PC\SharedFolder\ -- or -- <i>file:///Tests-PC\SharedFolder\</i>	A shared network folder where tests reside.
\Tests-PC\SharedFolder\MyTestNGTest.zip -- or -- <i>file:///Tests-PC\SharedFolder\MyTestNGTest.zip</i>	The archive (zip) file that contains the tests and resides in a shared network folder.
<i>http://downloads/mytests.com/MyTestNGTest.zip</i>	A web site URL path to the archive (zip) file that contains the tests files.

**Note:** If path you enter is incorrectly formatted, QAComplete will highlight the path in red.

- Alternately, upload your zipped file using **File Attachments**. Click **Browse** and navigate to the zipped test files.
- The **Run Mode** specifies how the test agent will get your test files and run the test.
  - To copy test files to the test host and run the test in a temporary folder there, choose **Copy the test project to and run it in a temporary folder** in the Run Mode drop-down list.
  - To run the test directly in the folder where the test files reside, choose **Run the test project in place (if possible)** from the Run Mode drop-down list. If the test files reside in a shared network folder or in a local folder on the test host, the test agent will run the test directly in that folder.
- If you want to run your test by using Maven, select **Use Maven Project**.
- Enter the test case class name in **Test Class**.  
If you use Maven to run tests, you can leave the field empty to run all test cases defined in your Selenium test.
- Click **Submit**.

## Assigning Selenium NUnit Tests to Tests

**Important:** You can use NUnit (Selenium) Test Agent to automate running of both Selenium NUnit tests and ordinary NUnit unit tests that do not use Selenium modules. The procedure of assigning tests to Test Library Tests is the same for both test types.

This instruction demonstrates how to assign a Selenium NUnit tests to a Test Library Test. Use the same procedure to assign an ordinary NUnit unit test to a Test.

### 1. Configure the Test Agent

On the Test Host that will run the test:

- Launch the Test Agent Settings tool (Start | All Programs | SmartBear | TestAgent | Test Agent Settings).
- Click **Advanced Settings**.
- If needed, specify the path to the NUnit executable (*nunit-console.exe*).

### 2. Prepare Your Selenium NUnit Test

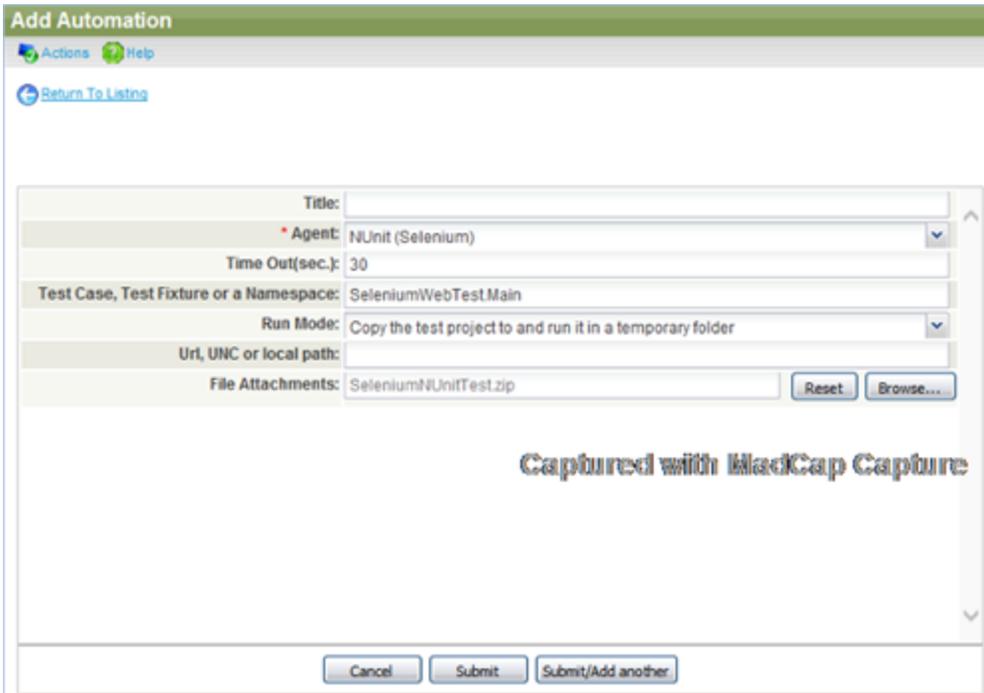
- If needed, zip up your Selenium NUnit test project (together with all libraries and files the project uses).
- Make note of where the project or the archive (zip file) resides.  
You can place the project (or the zip file containing the project) on a local folder on the test host where it will run or in a shared network folder. Make sure that the project can access all needed test files.  
Alternatively, you can upload the archive file to the server (see below).

### 3. Prepare Your Test Library Test

- In **Test Library**, open an existing Test for the NUnit test or add a new Test.
- Select a **Default Host Name**. If you skip this step, after you assign an automated test and click Submit, the system will remind you that the Test Host is missing.
- Save the changes.

#### 4. Assign Selenium NUnit Test to the Test

- On the **Automations** tab, click **Add New** to add an automated test.



- In the **Agent** drop-down list, select NUnit (Selenium).  
Note: If the Default Host assigned to the Test does not support the selected automated testing tool, the test will not run.
- You can leave the **Title** blank, and the system will use the name of the attached Selenium test, or you can enter a meaningful Title.
- The **Timeout** tells the Test Runner how long to let the test run. Enter the time out value in seconds. If the test takes longer than the given time out, the test runner will stop running.
- In **Url, UNC or local path**, enter the path to your NUnit project (.nunit) or to the archive (.zip) that contains your project.  
Below are examples of possible values:

Value	Description
C:\MyTests\MyNUnitTest.nunit -- or -- file:///C:\MyTests\MyNUnitTest.nunit	The NUnit project (.nunit) residing in the local folder on the test host.

Value	Description
C:\MyTests\ -- or -- file:///C:\MyTests\	The local folder on the test host where the NUnit project resides. Note: If there is more than one project in the folder, an error will occur.
C:\MyTests\MyNUnitTest.zip -- or -- file:///C:\MyTests\MyNUnitTest.zip	The archive (zip) file that contains the NUnit project and resides in a local folder on the test host.
\\Tests-PC\SharedFolder\MyNUnitTest.nunit -- or -- file:///Tests-PC\SharedFolder\MyNUnitTest.nunit	The NUnit project (.nunit) residing in a shared folder on a computer in your network.
\\Tests-PC\SharedFolder\ -- or -- file:///Tests-PC\SharedFolder\	A shared network folder where the NUnit project resides. Note: if there is more than one project in the folder, an error will occur.
\\Tests-PC\SharedFolder\MyNUnitTest.zip -- or -- file:///Tests-PC\SharedFolder\MyNUnitTest.zip	The archive (zip) file that contains the NUnit project and resides in a shared network folder.
http://downloads/mytests.com/MyNUnitTest.zip	A web site URL path to the archive (zip) file that contains the NUnit project.

**Note:** If path you enter is incorrectly formatted, QAComplete will highlight the path in red.

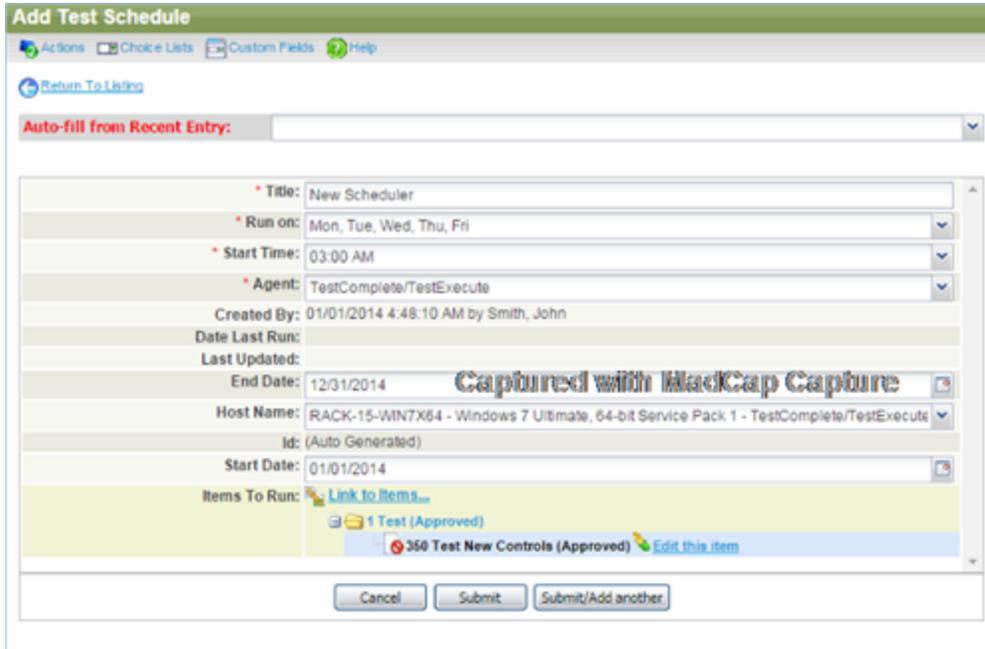
- Alternately, upload your zipped files using **File Attachments**. Click **Browse** and navigate to the zipped Selenium test files.
- The **Run Mode** specifies how the test agent will get your Selenium test files and run the test.
  - To copy test files to the test host and run the test in a temporary folder there, choose **Copy the test project to and run it in a temporary folder** in the Run Mode drop-down list.
  - To run the test directly in the folder where the test files reside, choose **Run the test project in place (if possible)** from the Run Mode drop-down list. If the test files reside in a shared network folder or in a local folder on the test host, the test agent will run the test directly in that folder.
- To run a specific test case, test fixture or a namespace, enter its full name in **Test Case, Test Fixture or Namespace**.

- Click **Submit**.

## Scheduling Automated Tests

You can configure Automated Tests or Test Sets to run independently of any user interaction by putting them on a Test Schedule. The Schedule can be a one-time scheduled run, or it can be a recurring schedule repeated on specific days of the week.

Select **Test Management > Test Schedules**.



The screenshot shows the 'Add Test Schedule' form with the following details:

- Title:** New Scheduler
- Run on:** Mon, Tue, Wed, Thu, Fri
- Start Time:** 03:00 AM
- Agent:** TestComplete/TestExecute
- Created By:** 01/01/2014 4:48:10 AM by Smith, John
- Date Last Run:**
- Last Updated:**
- End Date:** 12/31/2014
- Host Name:** RACK-15-WINTX64 - Windows 7 Ultimate, 64-bit Service Pack 1 - TestComplete/TestExecute
- Id:** (Auto Generated)
- Start Date:** 01/01/2014
- Items To Run:**
  - 1 Test (Approved)
  - 360 Test New Controls (Approved) [Edit this item](#)

On the **Test Schedules** listing form, click **Add New** to create a new schedule.

On the **Add Test Schedule** form, enter the information for the new schedule.

- In **Title**, give the schedule a meaningful title.
- In **Run On**, choose the days of the week to run the Automated Test.
- In **Start Time**, select the hour of the day when you want the Automated Test to run.
- Use the **Start Date** and **End Date** to define the date range that the schedule will be active. The schedule won't run until your start date, then will run as scheduled until the end date.

**Note:** If you leave the **Start Date** and **Empty Date** fields blank, their values will be the far past and the far future respectively.

- In the **Agent** list, select an automated testing tool you want to use to run tests:

- TestComplete/TestExecute
- Ready! API/SoapUI
- JUnit (Selenium)
- TestNG (Selenium)
- NUnit (Selenium)
- In the **Hosts** list, select the Test Host that will run the automated test. The list is populated with the machines where the Test Agent is installed and running. For more information on Test Agents and Hosts, please see [About Test Hosts](#) and [The Test Agent](#).
- In **Items to Run**, link your Automated Tests or Test Sets. These Tests or Test Sets will run on the Test Host in the order of selection.

Click **Submit** to save the schedule. At the scheduled time, the Test Agent will kick off the associated Test or Test Set on the Test Host selected for the Schedule.

**Note:** If the Tests are associated with a Test Host, the Host on the Schedule takes precedence: your test run on the Test Host on the Schedule, not the Host associated with the Test in the Test Library.

## Running Automated Tests

Now that you have installed and configured the Test Agent, linked it to a Configuration, associated an automated test with a Test Management Test and Test Host, you can run the Automated Test on demand from Test Management. Automated Tests and fully automated Test Sets can also be run unattended by using Test Schedules.

Like Manual Tests, you can run the Test directly from the Test Library or you can run the Test as part of a Test Set. Tests or Test Sets can be linked to the desired Release, Release/Iteration or Release/Iteration/Build and to the desired Configurations (for more information about Releases, please see About Releases. For more information on Configurations, please see Configurations.

- To run the Test as part of a Test Set, create a Test Set and add your automated Test to the Set. Click the **Run** button in the **Actions** column on the **Test Set List**.
  - The Test Set can be linked to your Release, Iteration or Build.
- To run the Test independently of a Test Set, you can click the **Run** button in the **Actions** column on the **Test Library List**.
  - A Test can also be linked to a Release, Iteration or Build.

The **Test Runner** window opens. If the Test or Test Set is linked to a Release, select the desired Release first. Select the Test Host in the **Run By Host** drop down, and click **Run**.

Your tests will run immediately on the associated Test Host. The Test Agent on the selected Test Host will perform the following steps:

- Connect to the QAComplete web service.
- Retrieve the automated tests associated with the Test.
- Start the appropriate automated testing tool (TestComplete/TestExecute, Ready! API, SoapUI or a unit testing framework).
- Run the automated test.

When complete, the results are posted as Run Results to the Test and/or Test Set.

### Important Notes:

- If you assign TestComplete automated tests to your Tests in Test Management, TestComplete or TestExecute must NOT be actively running on the Test Host when the Automated Test is run. Otherwise, the test will return an error. TestComplete/TestExecute will be automatically started up by the test runner.
- If the required testing tool is not installed on the Test Host or the Test Agent is not configured properly to run that testing tool, the test will fail.

- If your Test Host has an earlier version of Test Agent that does not support the assigned automated test, the test will not run.
- The Test Agent can be set to log off of Windows on the Host machine. This effectively closes any open applications, logs out, then logs back in with the credentials from the Test Agent configuration. There is an option on the configuration form to first check if that user is currently logged in, and avoid the log out process.

## Running Combined Tests

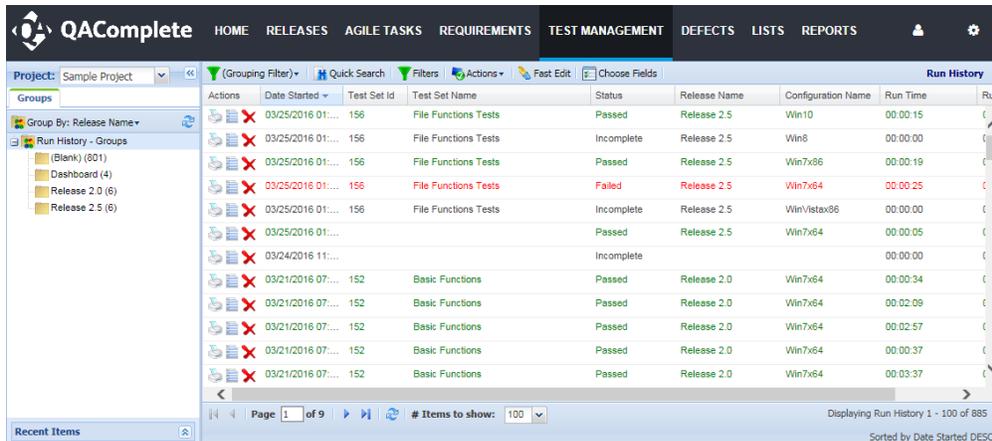
You can combine both manual and automated tests in the same test set. When you kick off the test runner, the automated tests are submitted on the associated Hosts, and you can continue running the manual tests as usual.

## Test Run History

You can view **test run history** from several different directions.

### Test Management / Run History

Go to the **Test Management > Run History** page. You will see a list of all runs, across all team members, test sets and tests:



Actions	Date Started	Test Set Id	Test Set Name	Status	Release Name	Configuration Name	Run Time	Run
	03/25/2016 01:...	156	File Functions Tests	Passed	Relesse 2.5	Win10	00:00:15	(
	03/25/2016 01:...	156	File Functions Tests	Incomplete	Relesse 2.5	Win8	00:00:00	(
	03/25/2016 01:...	156	File Functions Tests	Passed	Relesse 2.5	Win7x86	00:00:19	(
	03/25/2016 01:...	156	File Functions Tests	Failed	Relesse 2.5	Win7x64	00:00:25	(
	03/25/2016 01:...	156	File Functions Tests	Incomplete	Relesse 2.5	WinVistax86	00:00:00	(
	03/25/2016 01:...	156	File Functions Tests	Passed	Relesse 2.5	Win7x64	00:00:05	(
	03/24/2016 11:...	152	Basic Functions	Incomplete	Relesse 2.0	Win7x64	00:00:00	(
	03/21/2016 07:...	152	Basic Functions	Passed	Relesse 2.0	Win7x64	00:00:34	(
	03/21/2016 07:...	152	Basic Functions	Passed	Relesse 2.0	Win7x64	00:02:09	(
	03/21/2016 07:...	152	Basic Functions	Passed	Relesse 2.0	Win7x64	00:02:57	(
	03/21/2016 07:...	152	Basic Functions	Passed	Relesse 2.0	Win7x64	00:00:37	(
	03/21/2016 07:...	152	Basic Functions	Passed	Relesse 2.0	Win7x64	00:03:37	(

The test run history is color coded:

- **Green** for Passed
- **Red** for Failed
- **Black** for all other statuses

From here, you can drill into the details of each test run.

Click **Printer Friendly** to view a printable format of the run history.

Click **View Run** to see a read-only version of the Test Runner, where you can drill into each test in the run and into the test steps to view results.

When you delete test run history, the deletion will clean up and recalculate the Average Run Time for the tests and test sets involved. If you are deleting the most recent run for a Test Set, it will also clean up the Last Run information on the set and on the tests.

### Run History from the Test Library

To drill into run history for a specific test, you can go to the Test Library, open the test in Edit mode, then click the **Run History** tab.

Click **Printer Friendly** to view a printable format of the run history.

Click **View Run** to bring up read-only version of the Test Runner.

If the test was run as part of a test set, you will see the full view of the test set, where you can drill into each test in the run and into the test steps to view results.

#### Run History from the Test Set

To drill into run history for a specific test set, you can go to Test Sets, open the test set in Edit mode, then click the **Run History** tab.

Click **Printer Friendly** to view a printable format of the run history.

Click **View Run** to bring up read-only version of the Test Runner, where you can drill into each test in the run and then into the steps to view detailed results.

#### Run History - Automated Tests

When you submit automated tests, the test runner will post the top level of detail from the automated test result. From run history, you can drill into the detailed results:

- If you use TestComplete, Ready! API or SoapUI to run automated tests, the test runner will post the appropriate log file. Open the attached log file or download the file and view the detailed test results.  
**Note:** If you use [Test Agent](#) version 2.1, it exports TestComplete test logs to an HTML file. You can view them in any supported web browser. Otherwise, the test logs are stored in an MHT file. You can view them in Internet Explorer only.
- If you use NUnit to run Selenium automated tests or unit tests, the test runner will post NUnit test project log file. Download the file and view test results on your computer.
- If your Selenium automated test or unit test fails, the test runner will post the console output of the appropriate unit testing framework (JUnit, TestNG or NUnit) as Test Step Results.