Syncfusion®
Deliver innovation with ease®

Essential Studio 2013 Volume 4 - v.11.4.0.26

# Essential QuickTest Professional

# Contents

1.1 Introduction to Essential QuickTest Professional .... 4

1.2 Prerequisites and Compatibility .... 5

1.3 Documentation .... 6

**2 Installation and Deployment 7**

2.1 Installation and Configuration .... 7

2.1.1 Installation .... 7

2.1.2 Configuration .... 14

2.1.2.1 Automatic Configuration .... 14

2.1.2.2 Manual Configuration .... 15

2.2 Sample and Location .... 17

2.3 Licensing, Patches and Uninstallation .... 18

2.4 Assembly information .... 18

**3 Getting Started 22**

3.1 Creating and Recording a Test .... 22

3.2 Running a Test .... 33

3.3 Editing a Test .... 35

3.4 Saving a Test .... 39

3.5 Running the Saved Test .... 40

**4 Supported Controls and Methods 43**

4.1 Essential Grid .... 43

4.2 Essential Tools .... 51

4.3 Essential Chart .... 61

4.4 Essential Schedule .... 63

4.5 Essential Diagram .... 63

**5 Known Issues 65**

5.1 General .... 65

5.2 Essential Grid .... 65

5.3 Essential Tools .... 65

**6 Utilities 67**

6.1 Configuring the SwfConfig file ........ 67

**7 Frequently Asked Questions 71**

7.1 General ........ 71

7.1.1 How to manually configure Syncfusion control to work with QTP ........ 71

7.1.2 How to know whether my swfconfig file holds an invalid assembly path reference ........ 73

7.1.3 How to fetch installation information related to the Syncfusion QTP add-on ........ 76

7.1.4 Why are Syncfusion controls not recognized even after adding the custom libraries? ........ 78

7.1.5 How do I know that Essential QuickTest Professional works as expected? ........ 78

7.2 Essential Grid ........ 79

7.2.1 How to get the description of the Check Box Cells and Normal Cells in Essential Grid ........ 80

7.2.2 How to set the current cell in Grid ........ 81

7.3 Essential Tools ........ 82

7.3.1 How to select the XPtool bar without ID ........ 82

7.3.2 How to check and uncheck the CheckBoxAdv ........ 82

7.3.3 How to collapse and expand the specified node in TreeViewADV ........ 83

7.4 Essential Chart ........ 84

7.4.1 How to get the displayed text in the X-axis and Y-axis ........ 84

7.4.2 How to find the count of a series within the chart ........ 85

7.4.3 How to find the maximum Y-axis value in the specified region ........ 85

7.5 Essential Schedule ........ 86

7.5.1 How to reschedule the appointment to another timeline ........ 86

7.5.2 How to reschedule the timeline of an appointment ........ 87

7.6 Essential Diagram ........ 88

7.6.1 How to change the node to a new position ........ 88

7.6.2 How to connect the specified nodes using connectors ........ 88

7.6.3 How to resize the node ........ 89

# 1 Overview

This topic gives an introduction to the new Essential QuickTest Professional. It answers the following questions:

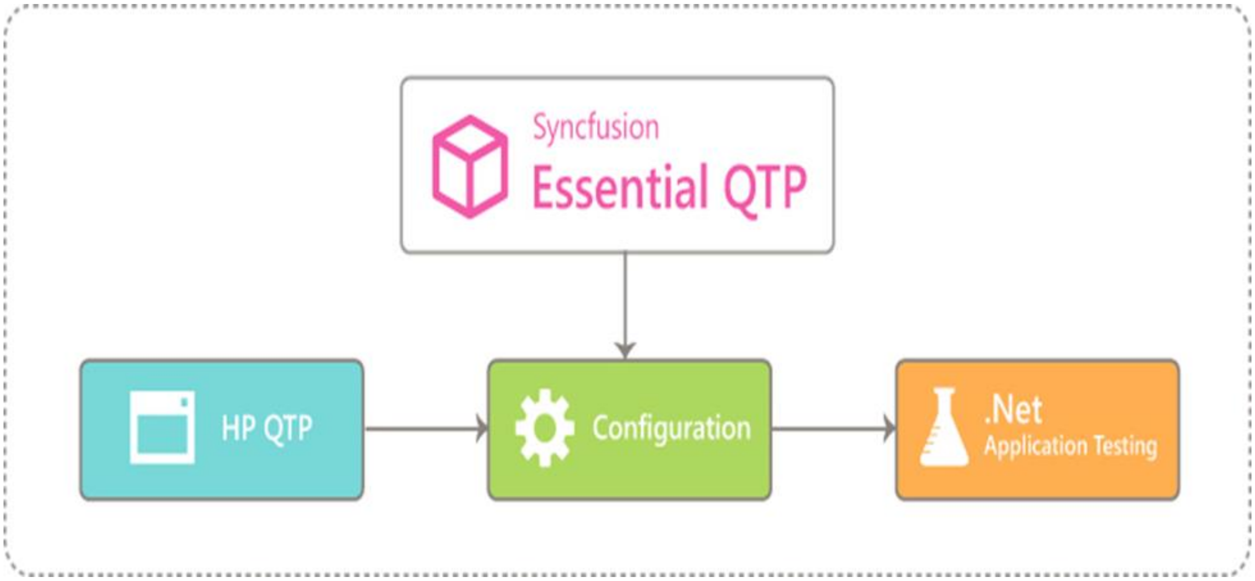
- What is Essential QuickTest Professional and how is it linked with QuickTest Professional (QTP)?
- Who can use the product or for whom is this product intended?
- What is the product used for?

You will also get an overview of what this manual has to offer.

## 1.1 Introduction to Essential QuickTest Professional

Essential QuickTest Professional is an add-on shipped with Essential Studio products offered by Syncfusion. It has been specially designed to meet the requirements of professionals who test the applications designed (using Syncfusion controls) with HP QuickTest Professional software.

Essential QuickTest Professional contains custom libraries, which help HP QuickTest Professional software recognize Syncfusion controls. These custom libraries are built with the help of QuickTest Professional .NET add-in extensibility. For more details, refer to HP QuickTest Professional help.



The custom libraries allow Syncfusion controls to be used as a native object inside the QTP testing environment and enable testing of applications in QTP. Essential QuickTest Professional will help you to perform regression test on your application containing Syncfusion controls and thereby increase the reliability of the end product. The following chapters demonstrate the usage of our custom library in QTP.

Essential QuickTest Professional comes with numerous samples as well as extensive documentation to guide you step-by-step. This user guide provides detailed information on the features and functionalities of ssential QuickTest Professional and is organized in the following order:

**Install and Configuration of Essential QuickTest Professional**

This section provides details about installing and configuring Essential QuickTest Professional, which is mandatory before you start using the add-on.

**Samples and Locations**

This section provides the location of the installed samples, source code location, and the location of the assemblies for the source.

**Licensing, Patches and Uninstallation**

This section covers information on licensing and patches. It also covers the uninstallation process.

**Assembly information**

This section provides details about the assembly name, type name, and control type in table format. This table is used to write the swfconfig file.

**Frequently Asked Questions**

This section comprises an assembled list of questions and answers to provide expert solutions on the product and its usage for every control that is supported.

**Known Issues**

This section lists existing issues with the product that have not yet been solved.

**Supported Controls**

Essential QuickTest Professional supports only certain methods exclusive to each control. This section lists these controls and their methods.

## 1.2 Prerequisites and Compatibility

This section covers the requirements that are mandatory for installing Essential Test Studio. It also lists the operating systems and browsers that are compatible with the product.

**Prerequisites**

The following are the prerequisites:

| | |
|---|---|
| **Testing Environments** | 1) QuickTest Professional version 9.5 and above<br>2) QuickTest Professional .NET add-in |

| | |
|---|---|
| **.NET Framework** | .Net Framework version 2.0, 3.5, 4.0, or 4.5 |
| **Other Requirements** | Essential Studio (User Interface edition – Windows Forms) of the same version as the Essential QTP Add-on. |

**Compatibility**

Essential QuickTest Professional is compatible with the following operating systems:

| | |
|---|---|
| **Operating Systems** | • Microsoft Windows XP<br>• Microsoft Windows Vista<br>• Microsoft Windows 7<br>• Microsoft Windows 8.1<br>• Microsoft Windows Server 2003<br>• Microsoft Windows Server 2008 |

## 1.3 Documentation

Syncfusion provides the following documentation segments which cover all the necessary information pertaining to Essential QuickTest Professional.

| Type of documentation | Location |
|---|---|
| Readme | (Installed location of Essential QTP)\ReadMe\ReadMe.htm |
| Class Reference | (Installed location of Essential QTP)\Help\ClassReference.chm |

# 2 Installation and Deployment

This section covers information on the install location, samples, licensing, and uninstallation of the recent version of Essential QuickTest Professional.

## 2.1 Installation and Configuration

This topic explains the installation process for Syncfusion Essential QuickTest professional and the configuration details about the swfconfig file.

### 2.1.1 Installation

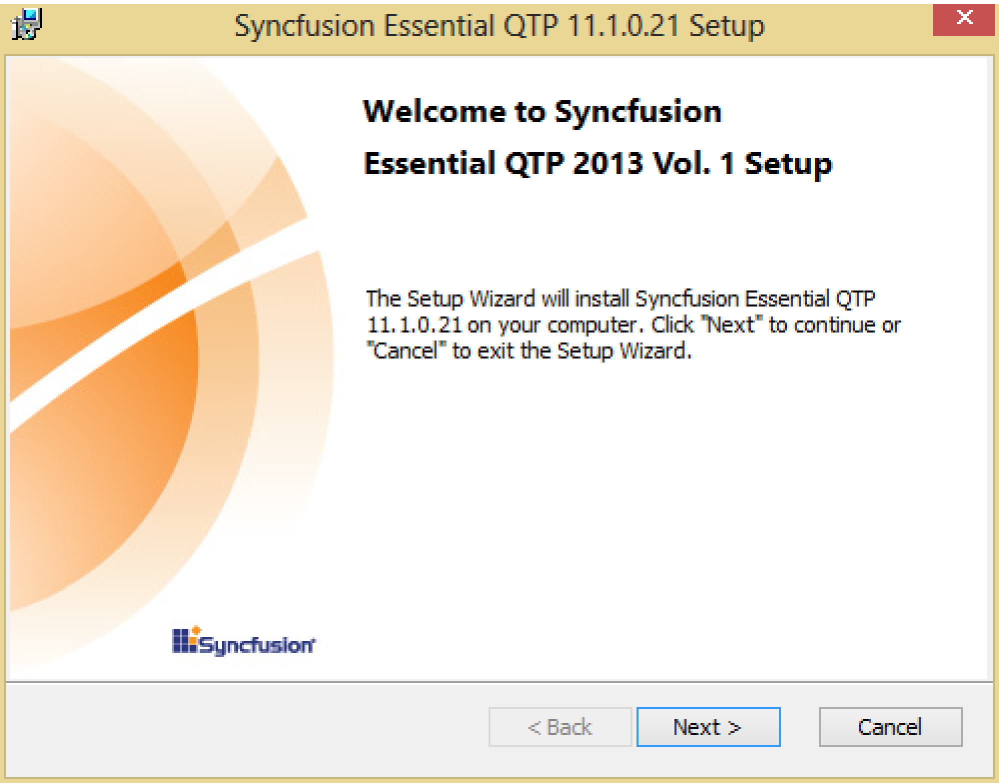
This section provides specifics on installation of Syncfusion Essential QuickTest professional.

To install Essential Test Studio:

***Note: The Installation procedures are the same for every Essential Test Studio setup, regardless of the volume of the Test setup.***

1. Double-click the Syncfusion Essential Test Studio Setup file.

***Note: Setup - Syncfusion Essential QuickTest Professional dialog box opens.***



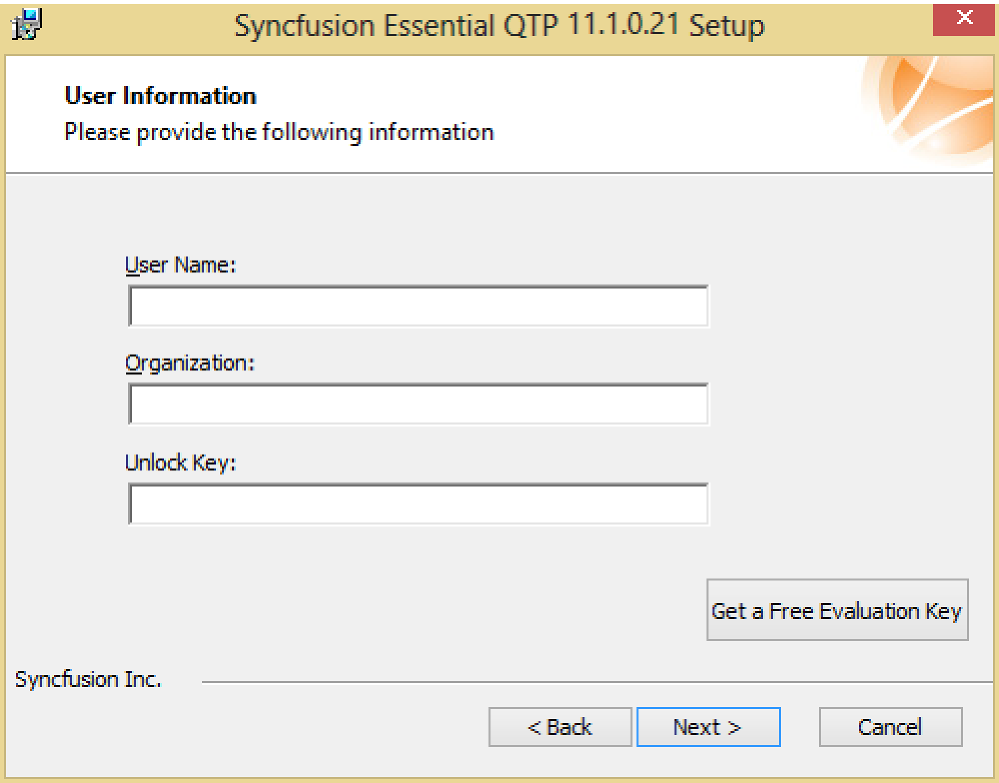
Syncfusion Essential QTP 11.1.0.21 Setup

Welcome to Syncfusion Essential QTP 2013 Vol. 1 Setup

The Setup Wizard will install Syncfusion Essential QTP 11.1.0.21 on your computer. Click "Next" to continue or "Cancel" to exit the Setup Wizard.

*Figure 1: Setup - Essential QuickTest Professional Welcome screen*

2. Click **Next**. The **User Information** screen opens.



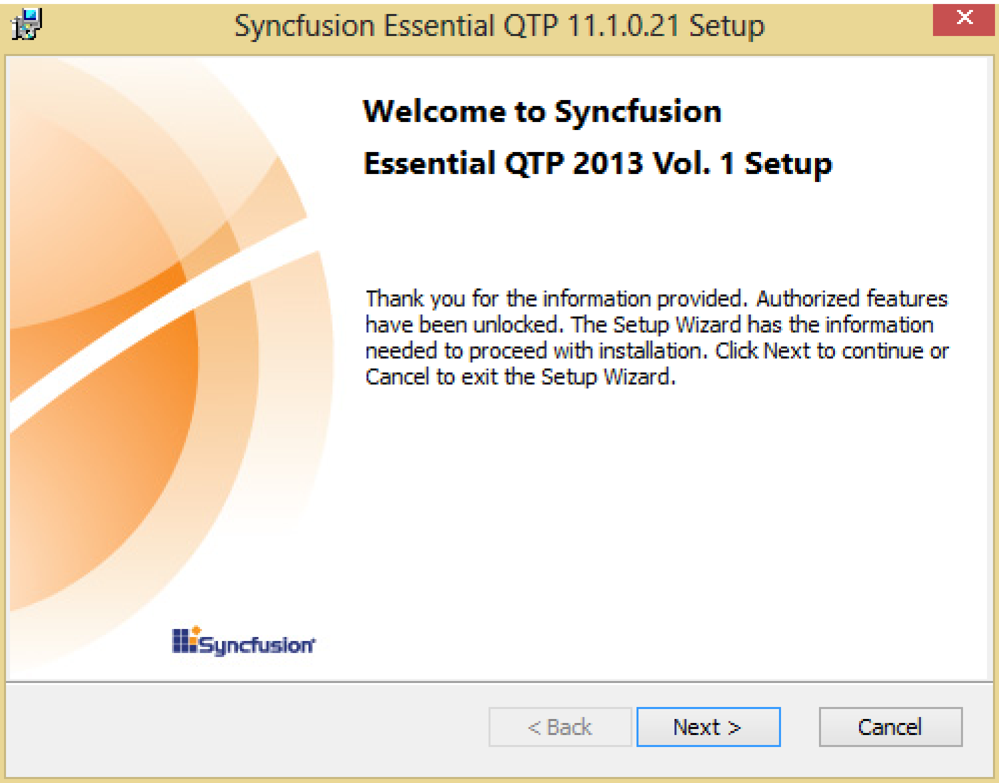
Syncfusion Essential QTP 11.1.0.21 Setup
User Information
Please provide the following information
User Name:
Organization:
Unlock Key:
Get a Free Evaluation Key
Syncfusion Inc.
< Back
Next >
Cancel

*Figure 2: User Information screen*

3. Enter your name, organization and enter the license key in the corresponding text boxes provided.

***Note: Use Essential Studio Unlock Key as the Unlock Key for Essential Testing Studio. For version previous to 6.3.0.6, use "Syncfusion199" as the Unlock Key.***

4. Click **Next**. The unlock key will be validated.



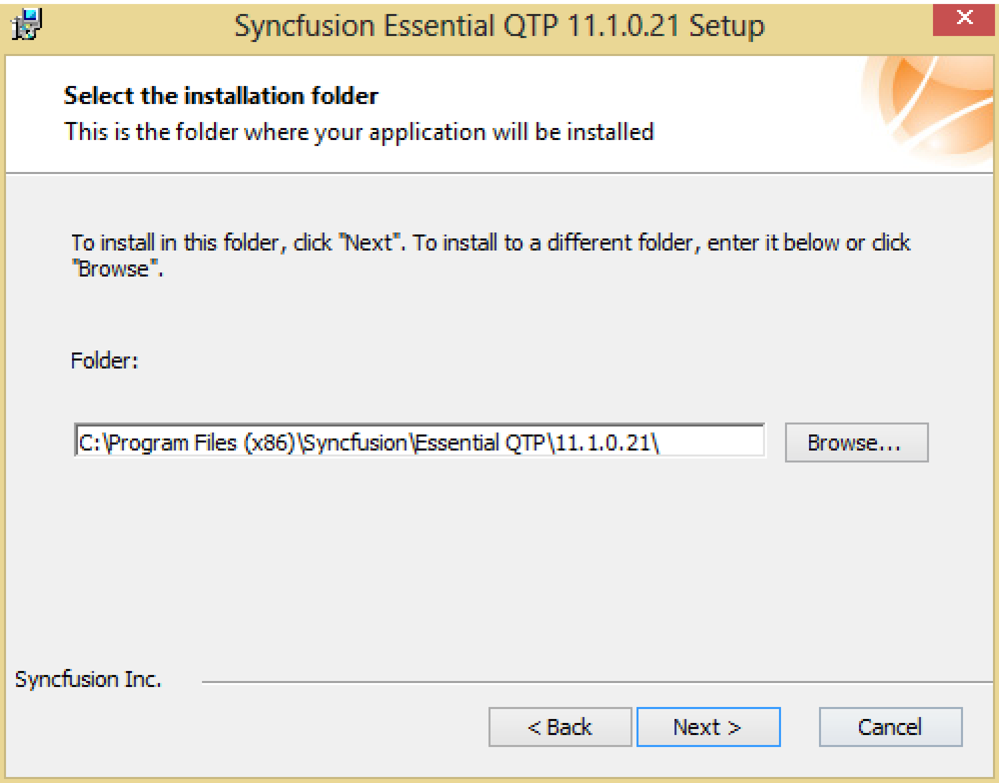
Syncfusion Essential QTP 11.1.0.21 Setup

Welcome to Syncfusion
Essential QTP 2013 Vol. 1 Setup

Thank you for the information provided. Authorized features have been unlocked. The Setup Wizard has the information needed to proceed with installation. Click Next to continue or Cancel to exit the Setup Wizard.

*Figure 3: Select Destination Location*

5. Click **Next**. The Select the installation folder window opens.



Syncfusion Essential QTP 11.1.0.21 Setup

**Select the installation folder**

This is the folder where your application will be installed

To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

Folder:

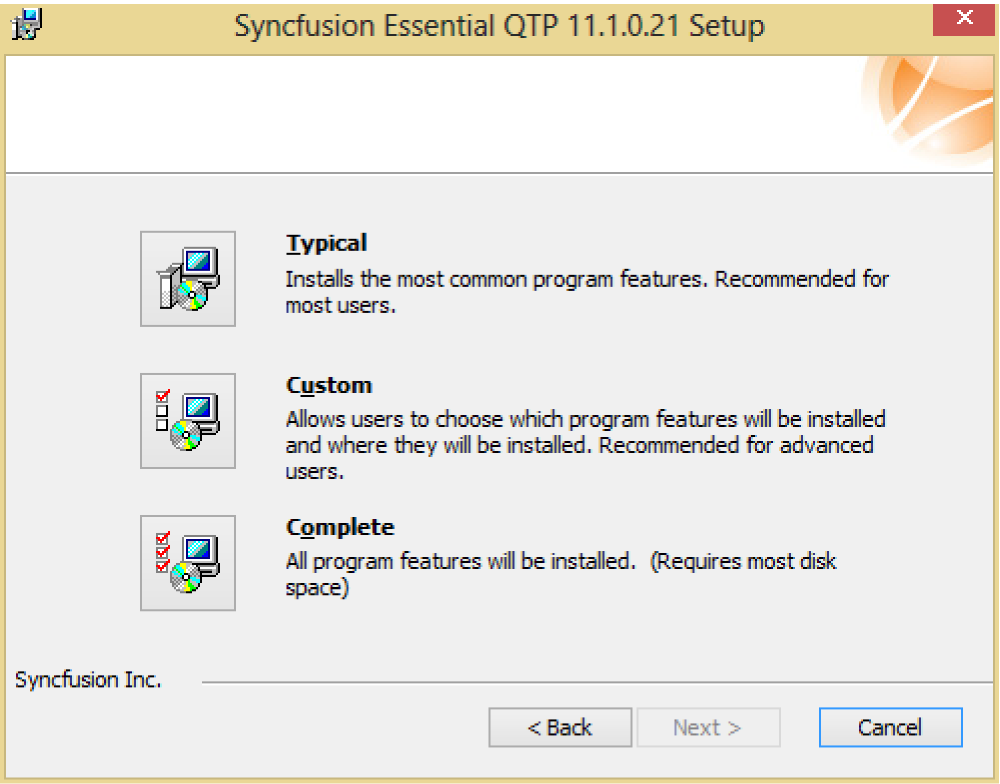
C:\Program Files (x86)\Syncfusion\Essential QTP\11.1.0.21\

*Figure 4: Select Installation Folder*

6. To install in the default location, click **Next**. You can also browse to choose a required location. When you click **Browse** to select the desired location, the Destination Location screen displays the selected location.

7. Click **Next**. The Installation options window opens. Choose one of the following installation options as required:

- **Typical** - Installs most common program features.
- **Custom** - Allows you to choose the program to be installed and where it should be installed.
- **Complete** - Installs all of the feature programs.



Syncfusion Essential QTP 11.1.0.21 Setup

**Typical**
Installs the most common program features. Recommended for most users.

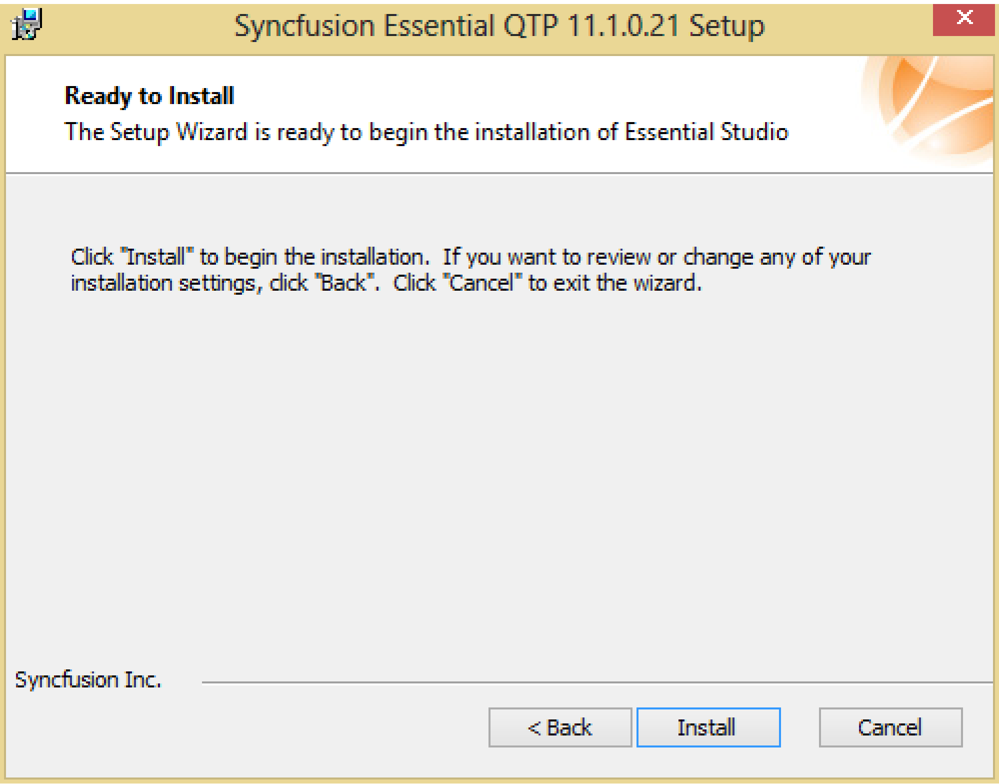
**Custom**
Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.

**Complete**
All program features will be installed. (Requires most disk space)

Syncfusion Inc.

*Figure 5: Installing options Screen*

8. Click **Next.** The **Ready to Install** dialog opens.



Syncfusion Essential QTP 11.1.0.21 Setup

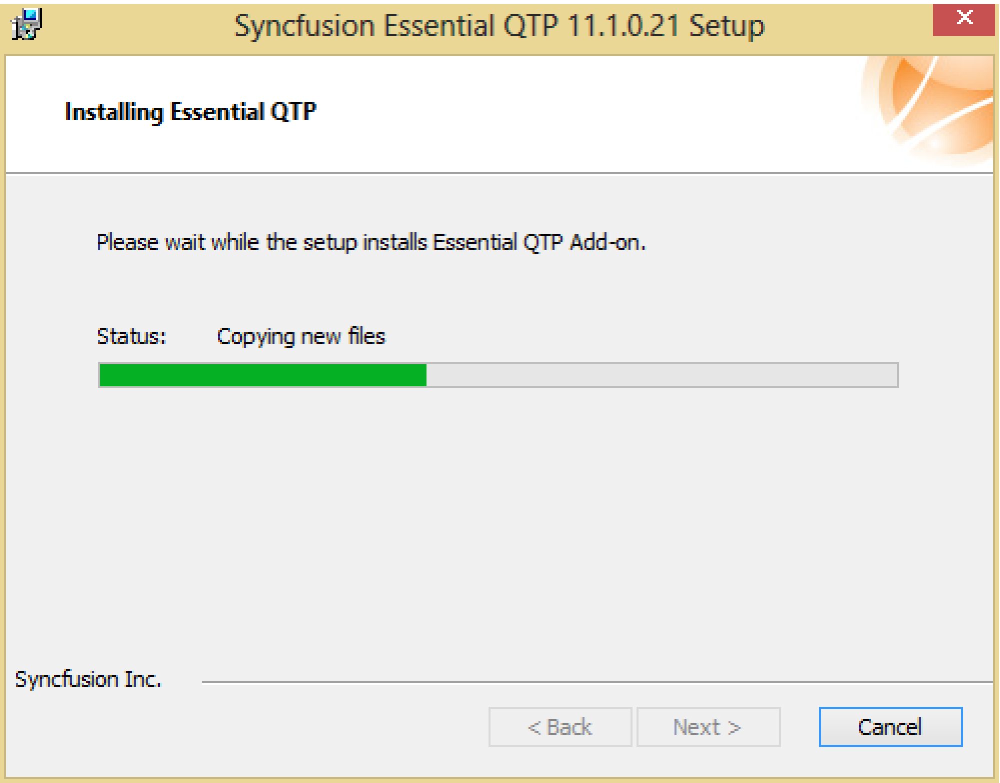
**Ready to Install**

The Setup Wizard is ready to begin the installation of Essential Studio

Click "Install" to begin the installation. If you want to review or change any of your installation settings, click "Back". Click "Cancel" to exit the wizard.

*Figure 6: Ready to Install Screen*

9. Click **Install**. The installation process starts displaying the Installing screen as shown in the following screenshot.



Syncfusion Essential QTP 11.1.0.21 Setup

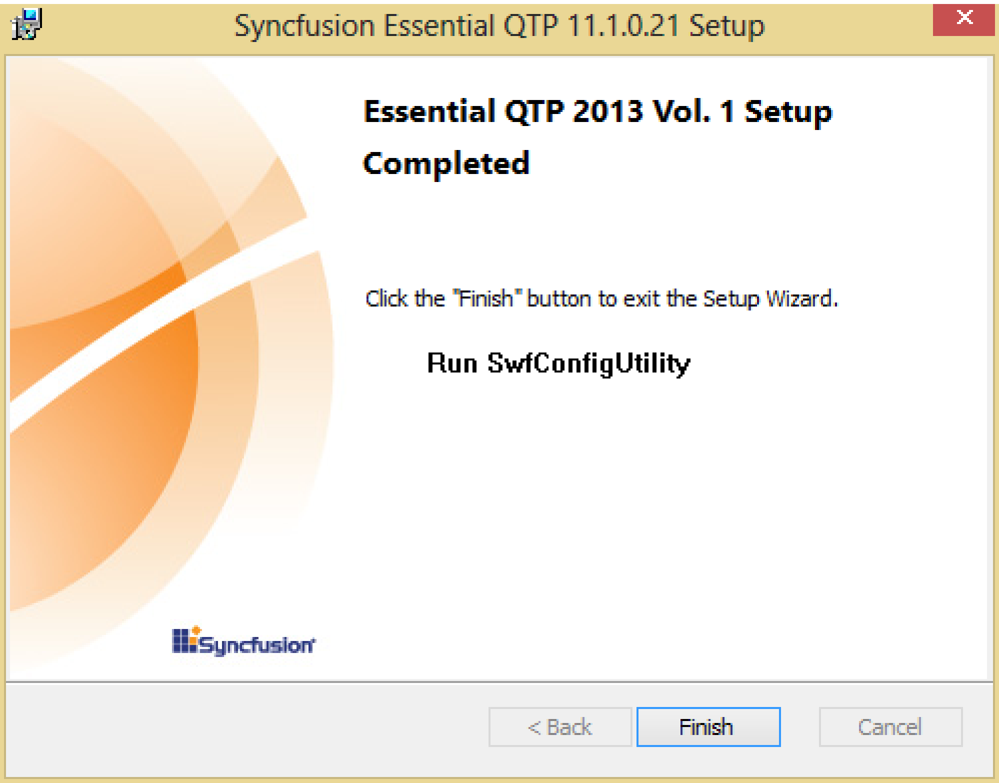
**Installing Essential QTP**

Please wait while the setup installs Essential QTP Add-on.

Status: Copying new files

*Figure 7: Installing Screen*

The following screen is displayed once the installation is completed.



Syncfusion Essential QTP 11.1.0.21 Setup

Essential QTP 2013 Vol. 1 Setup Completed

Click the "Finish" button to exit the Setup Wizard.

Run SwfConfigUtility

*Figure 8: Installation Completed*

### 2.1.2 Configuration

An XML file in QTP called **swfconfig** is the configuration file located at ***(Installed location of Essential QuickTest Professional)Config\<version-2.0, 3.5, or 4.0>\swfconfig***, which contains all the mapping information for QTP to recognize Syncfusion controls. In **swfconfig**, the controls are mapped to their corresponding custom server libraries (Essential QuickTest Professional DLLs) by giving the fully qualified name of the DLL.

***Note: The fully qualified name is the name of the file mentioned with its complete path.***

Any event that is triggered while working with a Syncfusion control, either by the user or the program activity, will be handled by the corresponding method in the custom library (DLL) given as the <DllName> tag under the <Control> tag.

An XML file can be configured in one of two ways, automatically or manually.

#### 2.1.2.1 Automatic Configuration

This section provides the details about the configuration of the swfconfig file using the SwfConfigUtility. Refer to the Utility section of this document.

### 2.1.2.2 Manual Configuration

This section provides details about the manual configuration of the swfconfig file.

**Steps to Configure QTP to use the Custom Libraries shipped in Essential QuickTest Professional**

1. Navigate to the following path:
   ***(Installed location of Essential QuickTest Professional)\Config***

***Note: You will find three folders, named 2.0, 3.5 and 4.0 here. The folders 2.0, 3.5 and 4.0 consist of swfconfig files for .NET 2.0, .NET 3.5 and NET 4.0 frameworks respectively.***

2. Open the swfconfig file by double-clicking it. You can view the mapping for all the supported controls here. The sample code below maps the grid control to its corresponding DLL.

```
    [XML]

<CC <Control Type="Syncfusion.Windows.Forms.Grid.GridControl">
<CustomRecord>
<Component>
 <Context>AUT</Context>
    <DllName>C:\Program files\Syncfusion\Essential TestStudio\<Version
    Number>\Bin\2.0\GridControl.dll</DllName>
    <TypeName>Syncfusion.TestStudio.Grid.GridControl</TypeName>
        </Component>
  </CustomRecord>
<CustomReplay>
<Component>
    <Context>AUT</Context>
 <DllName>C:\Program files\Syncfusion\Essential TestStudio\<Version
    number>\Bin\2.0\GridControl.dll</DllName>
 <TypeName>Syncfusion.TestStudio.Grid.GridControl</TypeName>
       </Component>
  </CustomReplay>
</Control>
```

***Note: In the preceding code, the fully qualified name of the DLL given in the <DllName> tag assumes that you have installed the Essential QuickTest Professional in the following default path:***

***C:\Program Files\Syncfusion\Essential QuickTest Professional\<Version number>***

If you have installed Essential QuickTest Professional in any other path, you need to give the correct path of the DLL in all the <DllName> tag. For example, if Essential QuickTest Professional is located in D:\Essential QuickTest Professional\<version number>, then the code should be as follows:

```
    [XML]

<CC <Control Type="Syncfusion.Windows.Forms.Grid.GridControl">
<CustomRecord>
<Component>
 <Context>AUT</Context>
 <DllName>D:\Essential TestStudio\<Version
    Number>\Bin\2.0\GridControl.dll</DllName>
 <TypeName>Syncfusion.TestStudio.Grid.GridControl</TypeName>
        </Component>
  </CustomRecord>
<CustomReplay>
<Component>
    <Context>AUT</Context>
 <DllName>D:\Essential TestStudio\<Version
    number>\Bin\2.0\GridControl.dll</DllName>
 <TypeName>Syncfusion.TestStudio.Grid.GridControl</TypeName>
       </Component>
  </CustomReplay>
</Control>
```

3. Select the segment of the code containing the controls to be tested.
4. On the **Edit** menu, click **Copy**.

***Note: While selecting the code for copying, exclude the following lines of code.***

```
[XML]

<?xml version="1.0" encoding="UTF-8" ?>
```

5. Open the SwfConfig.xml file located under the following location:
***<QuickTest Professional Installation Path>\dat\SwfConfig.xml***

6. Paste the copied segment under the <?xml> tag in SwfConfig.xml.

***Note: The SwfConfig.xml file will look like the following:***

```
    [XML]

    <?xml version="1.0" encoding="UTF-8" ?>
<Controls>
<CC   <Control Type="Syncfusion.Windows.Forms.Grid.GridControl">
<CustomRecord>
<Component>
 <Context>AUT</Context>
 <DllName>C:\Program files\Syncfusion\Essential TestStudio\<Version
    Number>\Bin\2.0\GridControl.dll</DllName>
 <TypeName>Syncfusion.TestStudio.Grid.GridControl</TypeName>
          </Component>
  </CustomRecord>
<CustomReplay>
<Component>
    <Context>AUT</Context>
 <DllName>C:\Program files\Syncfusion\Essential TestStudio\<Version
    number>\Bin\2.0\GridControl.dll</DllName>
 <TypeName>Syncfusion.TestStudio.Grid.GridControl</TypeName>
          </Component>
  </CustomReplay>
</Control>
      ..........
</Controls>
```

***Note: Ensure that the element <DllName> contains the correct path to the corresponding DLL.***

7. Save the SwfConfig.xml file.

8. Restart QTP once the SwfConfig.xml file is saved to refresh the mappings to the required controls, before starting the test.

***Note: Mapping for the required controls can be done in a similar manner.***

## 2.2 Sample and Location

This section contains the location of the samples, source code and assemblies.

**Samples Location**

The samples for Essential QuickTest Professional are available at the following locations:

- Grid samples- (installed location of the product)\Examples\Samples\Grid\
- Tools samples- (installed location of the product)\Examples\Samples\Tools\
- Chart samples- (installed location of the product)\Examples\Samples\Chart\
- Schedule samples- (installed location of the product)\Examples\Samples\Schedule\
- Diagram samples- (installed location of the product)\Examples\Samples\Diagram\

***Note: By default, the installed location of the product corresponds to- <Drive>:\Program Files\Syncfusion\Essential QuickTest Professional\<version number>***

The executable files for the samples are available under the following location:

- (Installed location of the product)\Examples\Samples\Bin\

***Note: There is no sample browser available to run the samples for Essential QuickTest Professional. You have to manually run the exe from the above-mentioned location.***

**Source CodeLocation**

The source code for Essential QuickTest Professional is available at the following location:

- (Installed location of the product)\Src\

**Assemblies Location**

The assemblies are available under the following location:

- (Installed location of the product)\Bin\

## 2.3 Licensing, Patches and Uninstallation

This section deals with license keys, patches and the uninstallation process.

**Licensing**

Essentialqtpaddonsetup is the setup file for Essential QuickTest Professional, which can be installed with the same license key that has been used to install Essential Studio. Essential QuickTest Professional does not require a separate license.

**Patches**

Patches are not provided for Essential QuickTest Professional. In case of any fix requested by the user, the assemblies are sent. These assemblies are then to be placed under the following location:

***(Installed location of the product)\Bin\***

**Uninstallation**

Uninstallation can be done with the help of the **unins000** file that is available in the installed location. Double-clicking the file uninstalls Essential QuickTest Professional.

## 2.4 Assembly information

The following table shows the assembly information for each of the controls supported by Essential QuickTest Professional.

**For Essential Grid**

| Assembly Name | Type Name | Control Name |
|---|---|---|
| GridControl.dll | Syncfusion.TestStudio.Grid.GridControl | Syncfusion.Windows.Forms.Grid.GridControl |
| GridDataBoundGrid.dll | Syncfusion.TestStudio.Grid.GridDataBoundGrid | Syncfusion.Windows.Forms.Grid. GridDataBoundGrid |
| GridGroupingControl.dll | Syncfusion.TestStudio.Grid.GridGroupingControl | Syncfusion.Windows.Forms.Grid. GridGroupingControl |
| GridListControl.dll | Syncfusion.TestStudio.Grid.GridListControl | Syncfusion.Windows.Forms.Grid. GridListControl |
| TabBarSplitterControl.dll | Syncfusion.TestStudio.Grid.TabBarControl | Syncfusion.Windows.Forms.TabBar |

**For Essential Tools**

| Assembly Name | Type Name | Control Name |
|---|---|---|
| RibbonControlAdv.dll | Syncfusion.TestStudio.Tools.RibbonControlAdv | Syncfusion.Windows.Forms.Tools.RibbonControlAdv |
| DockingManager.dll | Syncfusion.TestStudio.Tools.DockingManager | Syncfusion.Windows.Forms.Tools.DockingManager |
| XPMenus.dll | Syncfusion.TestStudio.Tools.XPMenus | Syncfusion.Windows.Forms.Tools.XPMenus.BarControlInternal |
| PopupMenu.dll | Syncfusion.TestStudio.Tools.XPMenuGrid | Syncfusion.Windows.Forms.Tools.XPMenus.MenuGrid |
| CommandBar.dll | Syncfusion.TestStudio.Tools.CommandBar | Syncfusion.Windows.Forms.Tools.XPMenus.CommandBarExt |
| XPToolBar.dll | Syncfusion.TestStudio.Tools.XPToolBar | Syncfusion.Windows.Forms.Tools.XPMenus.XPToolBar |
| TreeViewAdv.dll | Syncfusion.TestStudio.Tools.TreeViewAdv | Syncfusion.Windows.Forms.Tools.TreeViewAdv |
| CalculatorControl.dll | Syncfusion.TestStudio.Tools.CalculatorControl | Syncfusion.Windows.Forms.Tools.CalculatorControl |
| ProgressBarAdv.dll | Syncfusion.TestStudio.Tools.ProgressBarAdv | Syncfusion.Windows.Forms.Tools.ProgressBarAdv |

| | | |
|---|---|---|
| CheckBoxAdv.dll | Syncfusion.TestStudio.Tools.CheckBoxAdv | Syncfusion.Windows.Forms.Tools.CheckBoxAdv |
| RadioButtonAdv | Syncfusion.TestStudio.Tools.RadioButtonAdv | Syncfusion.Windows.Forms.Tools.RadioButtonAdv |
| ColorPickerUIAdv.dll | Syncfusion.TestStudio.Tools.ColorPickerUI | Syncfusion.Windows.Forms.Tools.ColorPickerUIAdv |
| DateTimePickerAdv.dll | Syncfusion.TestStudio.Tools.DateTimePickerAdv | Syncfusion.Windows.Forms.Tools.DateTimePickerAdv |
| ThemedCheckButton.dll | Syncfusion.TestStudio.Tools.ThemedCheckButton | Syncfusion.Windows.Forms.ThemedCheckButton |
| ButtonAdv.dll | Syncfusion.TestStudio.Tools.ButtonAdv | Syncfusion.Windows.Forms.ButtonAdv |
| TextBoxExt.dll | Syncfusion.TestStudio.Tools.TextBoxExt | Syncfusion.Windows.Forms.Tools.TextBoxExt |
| MultiColumnComboBox.dll | Syncfusion.TestStudio.Tools.MultiColumnComboBox | Syncfusion.Windows.Forms.Tools.MultiColumnComboBox |
| TabControlAdv.dll | Syncfusion.TestStudio.Tools.TabControlAdv | Syncfusion.Windows.Forms.Tools.TabControlAdv |
| ScrollerFrame.dll | Syncfusion.TestStudio.Tools.ScrollerFrame | Syncfusion.Windows.Forms.ScrollBarCustomDraw |
| GroupBar.dll | Syncfusion.TestStudio.Tools.GroupBar | Syncfusion.Windows.Forms.Tools.GroupBar |
| GroupView.dll | Syncfusion.TestStudio.Tools.GroupView | Syncfusion.Windows.Forms.Tools.GroupView |
| TaskBarBox.dll | Syncfusion.TestStudio.Tools.TaskBarBox | Syncfusion.Windows.Forms.Tools.XPTaskBarBox |
| ComboDropDown.dll | Syncfusion.TestStudio.Tools.ComboDropDown | Syncfusion.Windows.Forms.Tools.ComboDropDown |
| DataListView.dll | Syncfusion.TestStudio.Tools.DataListView | Syncfusion.Windows.Forms.Tools.DataListView |
| ComboBoxAutoComplete.dll | Syncfusion.TestStudio.Tools.ComboBoxAutoComplete | Syncfusion.Windows.Forms.Tools.ComboBoxAutoComplete |
| TabbedMDI.dll | Syncfusion.TestStudio.Tools.TabbedMDI | Syncfusion.Windows.Forms.Tools.MDITabPanel |

### For Essential Chart

| Assembly Name | Type Name | Control Name |
|---|---|---|
| ChartControl.dll | Syncfusion.TestStudio. Chart.ChartControl | Syncfusion.Windows.Forms.Chart.ChartControl |

### For Essential Schedule

| Assembly Name | Type Name | Control Name |
|---|---|---|
| ScheduleControl.dll | Syncfusion.TestStudio Schedule.ScheduleControl | Syncfusion.Windows.Forms. Schedule.ScheduleControl |

### For Essential Diagram

| Assembly Name | Type Name | Control Name |
|---|---|---|
| Diagram.dll | Syncfusion.TestStudio.Diagram.Diagram | Syncfusion.Windows.Forms.Diagram.Controls.Diagram |

# 3 Getting Started

Essential QuickTest Professional lets you test applications with different Syncfusion controls, and allows playback of scripts. The following is a list of chapters containing information on the functionality of this software.

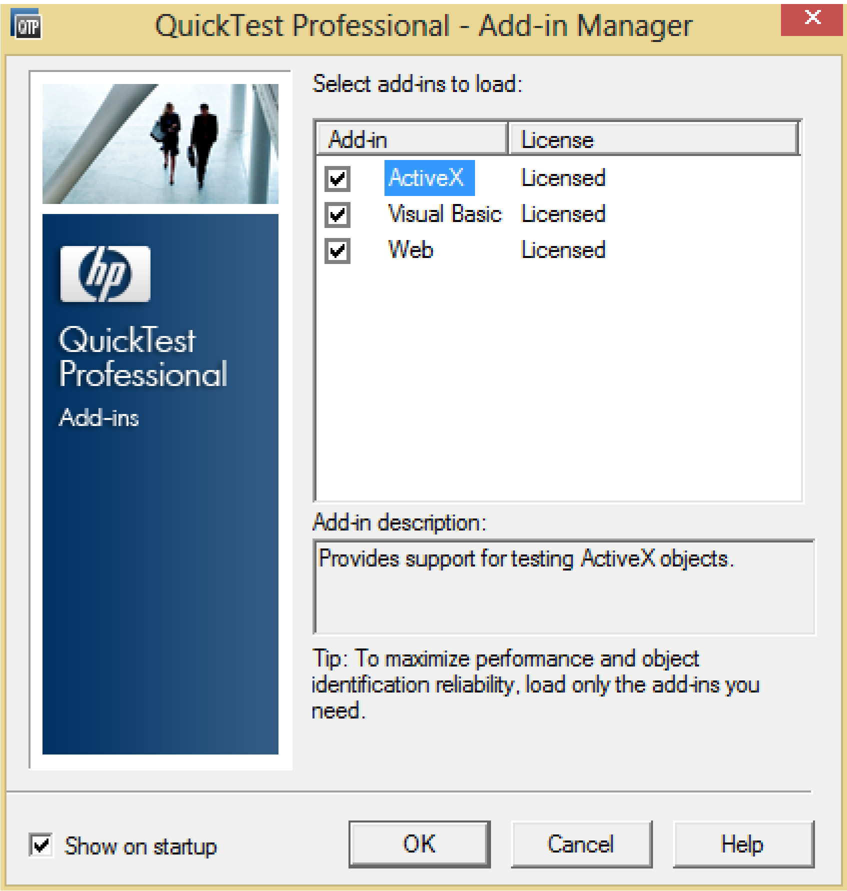
## 3.1 Creating and Recording a Test

To create a new test:

1. Open QTP by double-clicking the QuickTest Professional icon.

***Note: The QuickTest Professional – Add-in Manager window is displayed.***

2. Select the .NET check box under the **Add-in** header. This ensures that .NET add-in is installed.



QuickTest Professional - Add-in Manager

Select add-ins to load:

| Add-in | License |
|---|---|
| ☑ ActiveX | Licensed |
| ☑ Visual Basic | Licensed |
| ☑ Web | Licensed |

Add-in description:

Provides support for testing ActiveX objects.

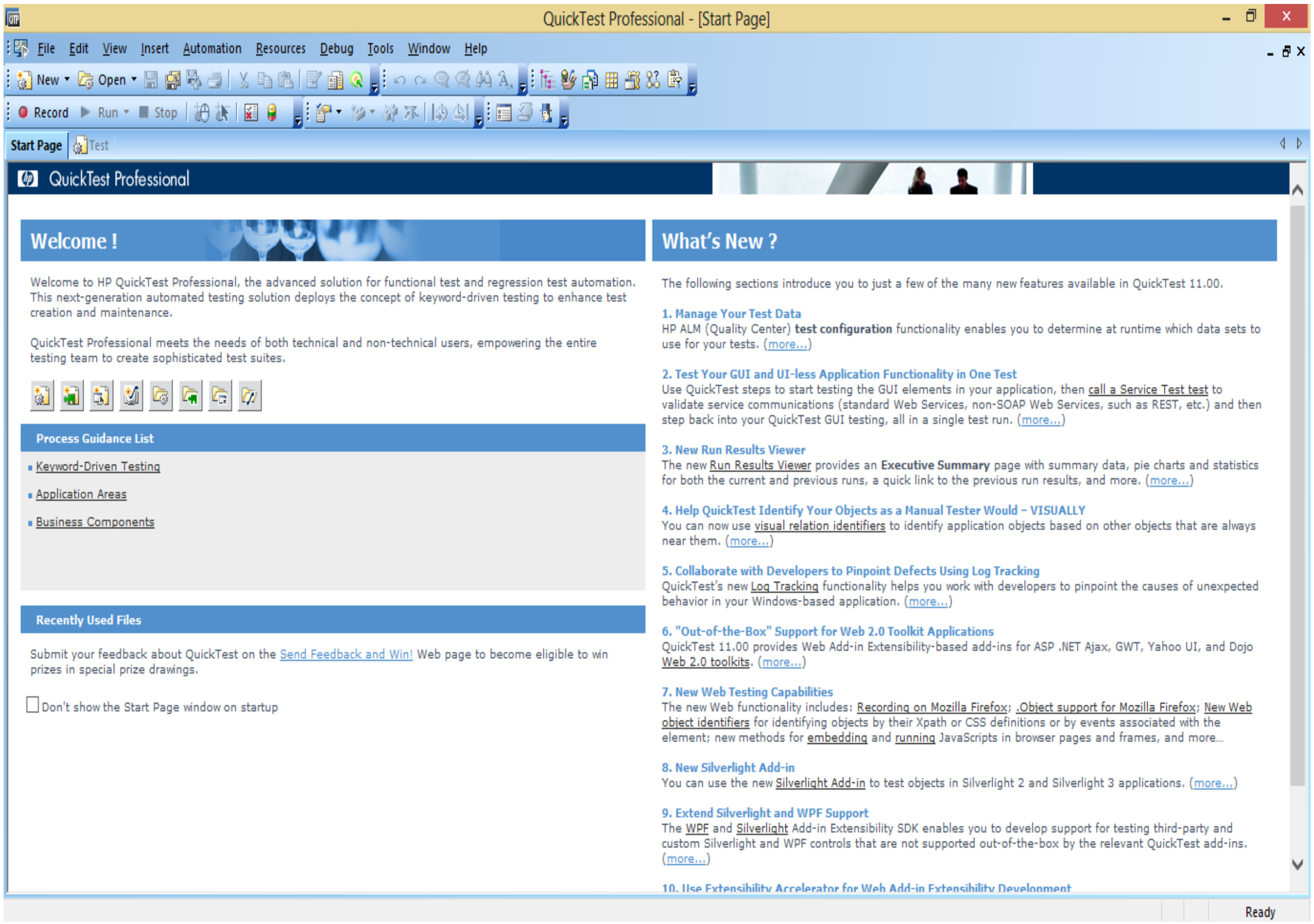
Tip: To maximize performance and object identification reliability, load only the add-ins you need.

☑ Show on startup

*Figure 9: QuickTest Professional - Add-In Manager*

3. Click **OK**.

***Note:The QuickTest Professional – [Start Page] window is displayed. There are two tabs, Start Page and Test, in the main pane of the window. The content under the Start Page tab is displayed by default.***


QuickTest Professional - [Start Page]
File Edit View Insert Automation Resources Debug Tools Window Help
New Open
Record Run Stop
Start Page
Test
QuickTest Professional

Welcome !

Welcome to HP QuickTest Professional, the advanced solution for functional test and regression test automation. This next-generation automated testing solution deploys the concept of keyword-driven testing to enhance test creation and maintenance.

QuickTest Professional meets the needs of both technical and non-technical users, empowering the entire testing team to create sophisticated test suites.

Process Guidance List

- Keyword-Driven Testing
- Application Areas
- Business Components

Recently Used Files

Submit your feedback about QuickTest on the Send Feedback and Win! Web page to become eligible to win prizes in special prize drawings.

□ Don't show the Start Page window on startup

What's New ?

The following sections introduce you to just a few of the many new features available in QuickTest 11.00.

1. Manage Your Test Data
HP ALM (Quality Center) **test configuration** functionality enables you to determine at runtime which data sets to use for your tests. (more...)

2. Test Your GUI and UI-less Application Functionality in One Test
Use QuickTest steps to start testing the GUI elements in your application, then call a Service Test test to validate service communications (standard Web Services, non-SOAP Web Services, such as REST, etc.) and then step back into your QuickTest GUI testing, all in a single test run. (more...)

3. New Run Results Viewer
The new Run Results Viewer provides an **Executive Summary** page with summary data, pie charts and statistics for both the current and previous runs, a quick link to the previous run results, and more. (more...)

4. Help QuickTest Identify Your Objects as a Manual Tester Would – VISUALLY
You can now use visual relation identifiers to identify application objects based on other objects that are always near them. (more...)

5. Collaborate with Developers to Pinpoint Defects Using Log Tracking
QuickTest's new Log Tracking functionality helps you work with developers to pinpoint the causes of unexpected behavior in your Windows-based application. (more...)

6. "Out-of-the-Box" Support for Web 2.0 Toolkit Applications
QuickTest 11.00 provides Web Add-in Extensibility-based add-ins for ASP .NET Ajax, GWT, Yahoo UI, and Dojo Web 2.0 toolkits. (more...)

7. New Web Testing Capabilities
The new Web functionality includes: Recording on Mozilla Firefox; .Object support for Mozilla Firefox; New Web object identifiers for identifying objects by their Xpath or CSS definitions or by events associated with the element; new methods for embedding and running JavaScripts in browser pages and frames, and more...

8. New Silverlight Add-in
You can use the new Silverlight Add-in to test objects in Silverlight 2 and Silverlight 3 applications. (more...)

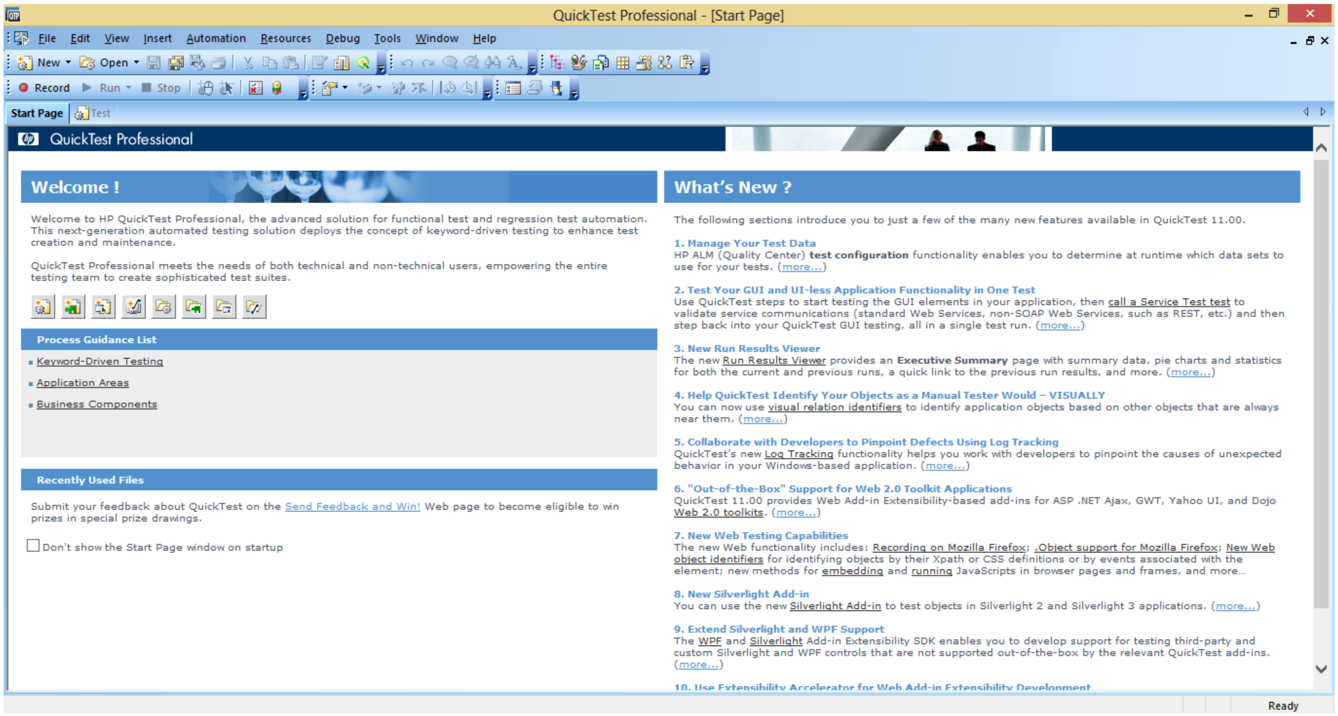
9. Extend Silverlight and WPF Support
The WPF and Silverlight Add-in Extensibility SDK enables you to develop support for testing third-party and custom Silverlight and WPF controls that are not supported out-of-the-box by the relevant QuickTest add-ins. (more...)

10. Use Extensibility Accelerator for Web Add-in Extensibility Development

Ready

*Figure 10: QuickTest Professional – [Start Page]*

4. Click the **New Test** icon on the Start Page.


QuickTest Professional - [Start Page]
File Edit View Insert Automation Resources Debug Tools Window Help
New Open
Record Run Stop
Start Page Test
QuickTest Professional
Ready

Welcome !

Welcome to HP QuickTest Professional, the advanced solution for functional test and regression test automation. This next-generation automated testing solution deploys the concept of keyword-driven testing to enhance test creation and maintenance.

QuickTest Professional meets the needs of both technical and non-technical users, empowering the entire testing team to create sophisticated test suites.

Process Guidance List

- Keyword-Driven Testing
- Application Areas
- Business Components

Recently Used Files

Submit your feedback about QuickTest on the Send Feedback and Win! Web page to become eligible to win prizes in special prize drawings.

□ Don't show the Start Page window on startup

What's New ?

The following sections introduce you to just a few of the many new features available in QuickTest 11.00.

1. Manage Your Test Data
HP ALM (Quality Center) **test configuration** functionality enables you to determine at runtime which data sets to use for your tests. (more...)

2. Test Your GUI and UI-less Application Functionality in One Test
Use QuickTest steps to start testing the GUI elements in your application, then call a Service Test test to validate service communications (standard Web Services, non-SOAP Web Services, such as REST, etc.) and then step back into your QuickTest GUI testing, all in a single test run. (more...)

3. New Run Results Viewer
The new Run Results Viewer provides an **Executive Summary** page with summary data, pie charts and statistics for both the current and previous runs, a quick link to the previous run results, and more. (more...)

4. Help QuickTest Identify Your Objects as a Manual Tester Would – VISUALLY
You can now use visual relation identifiers to identify application objects based on other objects that are always near them. (more...)

5. Collaborate with Developers to Pinpoint Defects Using Log Tracking
QuickTest's new Log Tracking functionality helps you work with developers to pinpoint the causes of unexpected behavior in your Windows-based application. (more...)

6. "Out-of-the-Box" Support for Web 2.0 Toolkit Applications
QuickTest 11.00 provides Web Add-in Extensibility-based add-ins for ASP .NET Ajax, GWT, Yahoo UI, and Dojo Web 2.0 toolkits. (more...)

7. New Web Testing Capabilities
The new Web functionality includes: Recording on Mozilla Firefox; .Object support for Mozilla Firefox; New Web object identifiers for identifying objects by their Xpath or CSS definitions or by events associated with the element; new methods for embedding and running JavaScripts in browser pages and frames, and more...

8. New Silverlight Add-in
You can use the new Silverlight Add-in to test objects in Silverlight 2 and Silverlight 3 applications. (more...)

9. Extend Silverlight and WPF Support
The WPF and Silverlight Add-in Extensibility SDK enables you to develop support for testing third-party and custom Silverlight and WPF controls that are not supported out-of-the-box by the relevant QuickTest add-ins. (more...)

10. Use Extensibility Accelerator for Web Add-in Extensibility Development

*Figure 11: QuickTest Professional – [Start Page] showing New Test icon*

This creates a new test. Alternatively, you can click the **Test** tab in the main pane of the window or Test sub-menu under the **New** menu in the **Menu** bar.

5. Click **Record** in the toolbar to start the recording.

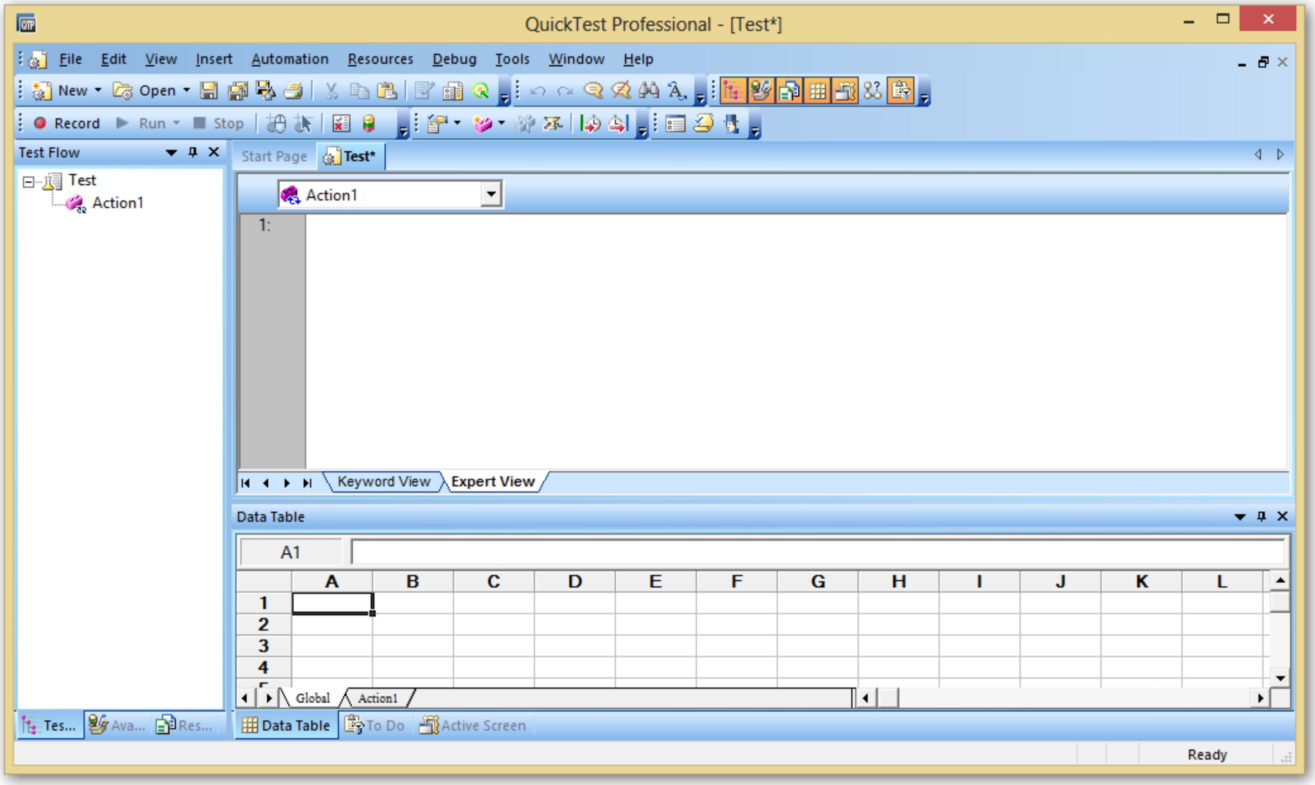
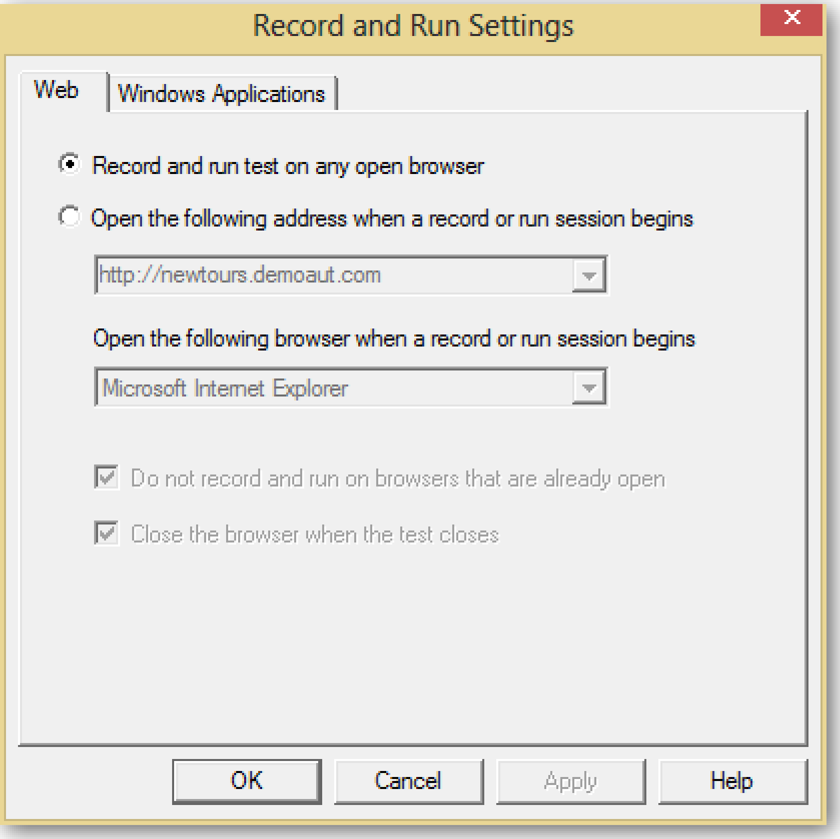


*Figure 12: QuickTest Professional – [Test*] Window showing Record tool*

***Note: A Record and Run Settings dialog box is displayed.***



Record and Run Settings

● Record and run test on any open browser

○ Open the following address when a record or run session begins

http://newtours.demoaut.com

Open the following browser when a record or run session begins

Microsoft Internet Explorer

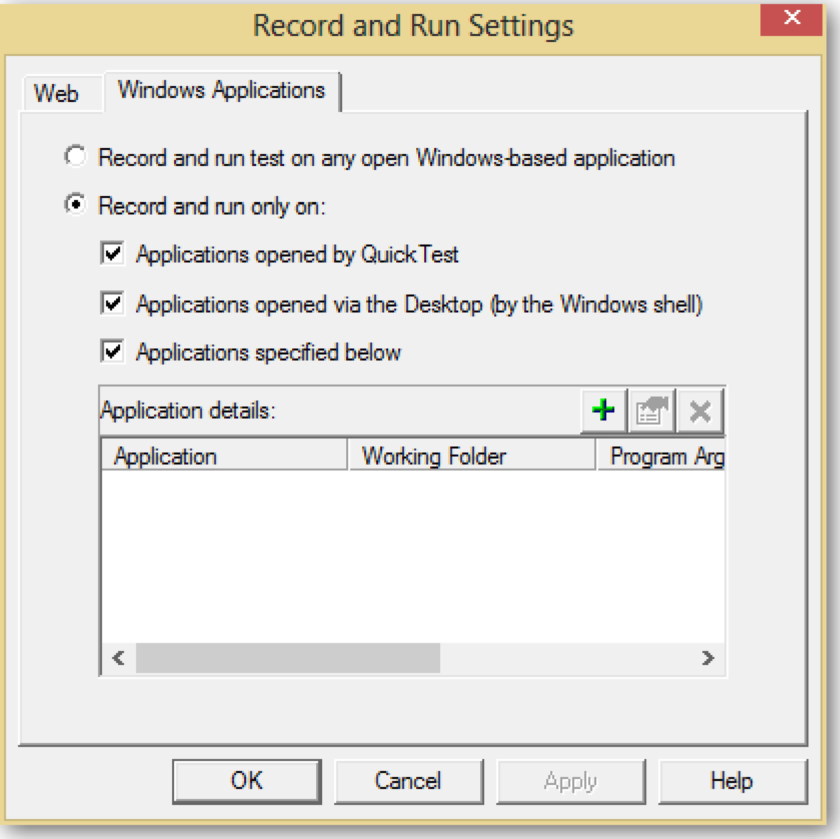
☑ Do not record and run on browsers that are already open

☑ Close the browser when the test closes

*Figure 13: Record and Run Settings-Web tab*

6. Click the Windows Applications tab.

***Note: The content under the tab is displayed.***



Record and Run Settings

○ Record and run test on any open Windows-based application

● Record and run only on:

☑ Applications opened by QuickTest

☑ Applications opened via the Desktop (by the Windows shell)

☑ Applications specified below

Application details:

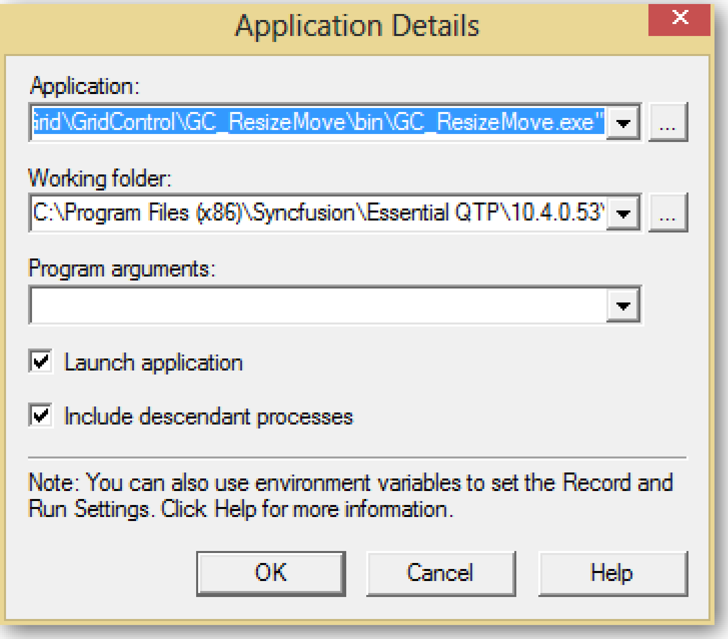
| Application | Working Folder | Program Arg |
|---|---|---|

*Figure 14: Record and Run Settings-Windows Applications*

***Note: The Record and run only on option button is selected by default, and the check boxes selected under it ensure that only the applications opened by QuickTest and added applications are tested.***

7. To add an application for testing, click the + button in the **Application details:** frame as shown in the figure above.

***Note: The Application Details dialog box is displayed.***



Application: rid\GridControl\GC_ResizeMove\bin\GC_ResizeMove.exe"

Working folder: C:\Program Files (x86)\Syncfusion\Essential QTP\10.4.0.53'

Program arguments:

☑ Launch application

☑ Include descendant processes

Note: You can also use environment variables to set the Record and Run Settings. Click Help for more information.

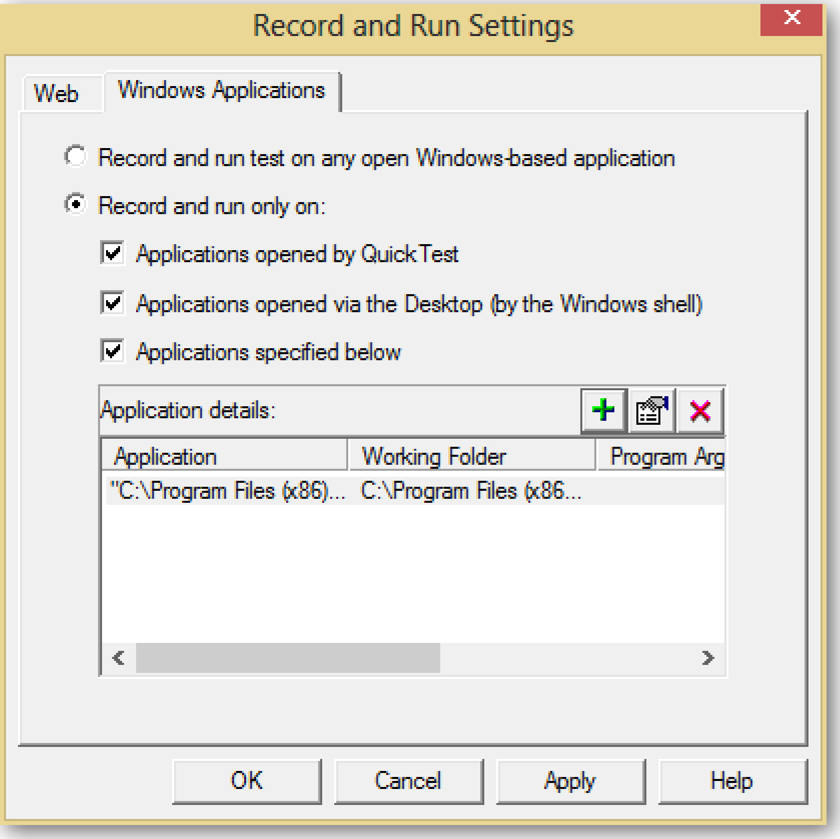
*Figure 15: Application Details*

8. Browse and select the path of the application that is to be tested by clicking (...) for the **Application:** label.
9. Browse and select the path of the working folder by clicking (...) for the **Working folder:** label.

***Note: Selecting the Launch application check box launches the application immediately after clicking OK in the current dialog. The Include descendant processes check box includes all the processes that are descendant to the current process. Both these check boxes will be selected by default.***

10. Click **OK**.

***Note: The path of the application and working folder are displayed in the Application details frame as shown below.***



Record and Run Settings

Web | Windows Applications

○ Record and run test on any open Windows-based application

● Record and run only on:

☑ Applications opened by QuickTest

☑ Applications opened via the Desktop (by the Windows shell)

☑ Applications specified below

Application details:

| Application | Working Folder | Program Arg |
|---|---|---|
| "C:\Program Files (x86)... | C:\Program Files (x86... | |

*Figure 16: Record and Run Settings with the Application Location*

11. Click **OK**.

***Note: The recording starts. The application in the given path is opened as shown below.***

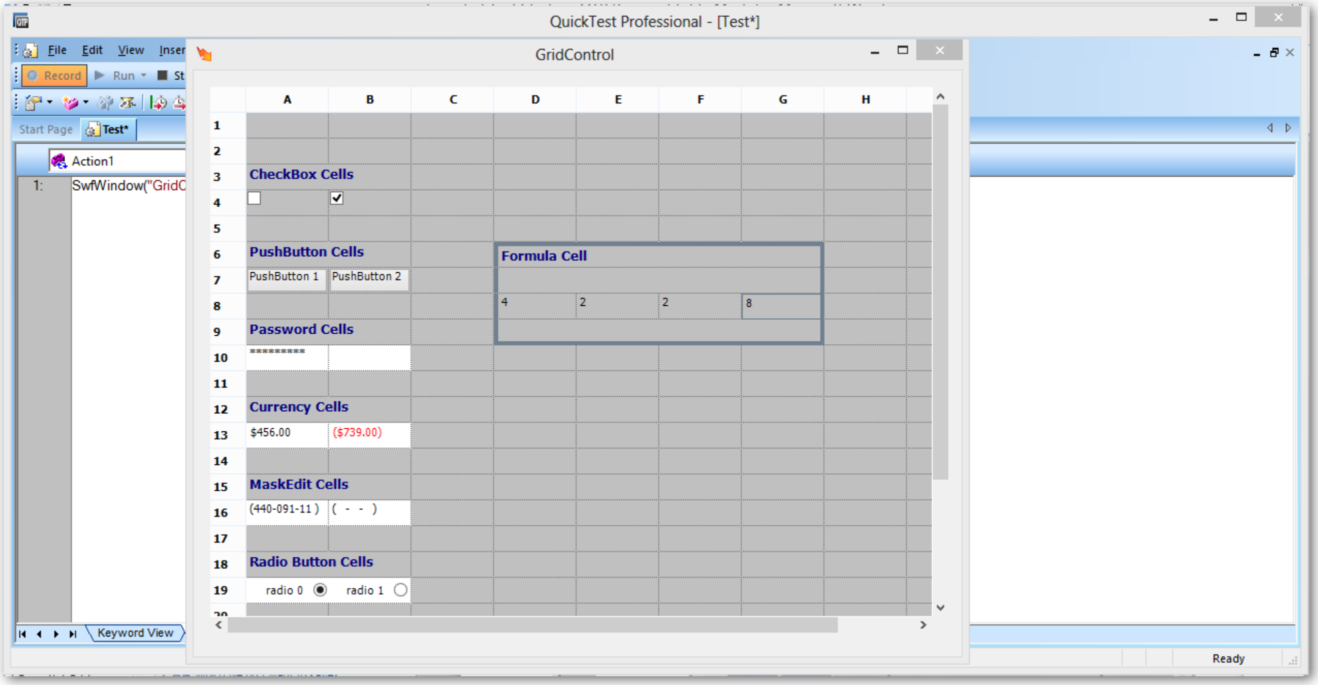


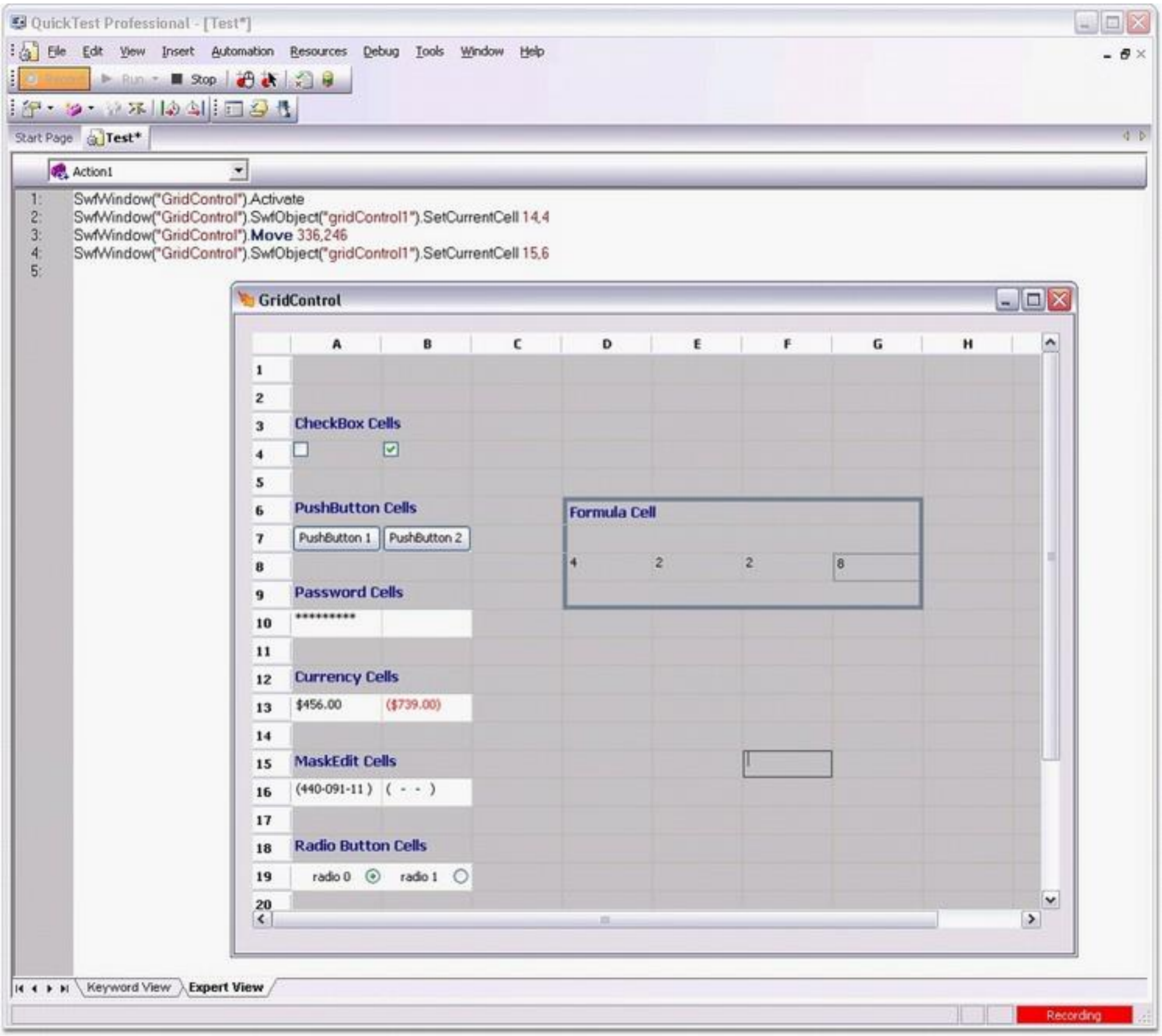
Figure 17: Application using Grid Control

1. Perform required valid user-action in the application.

***Note: Whenever the user performs any action involving the Syncfusion control used in the application, the SwfConfig file maps the control to the corresponding DLL.***

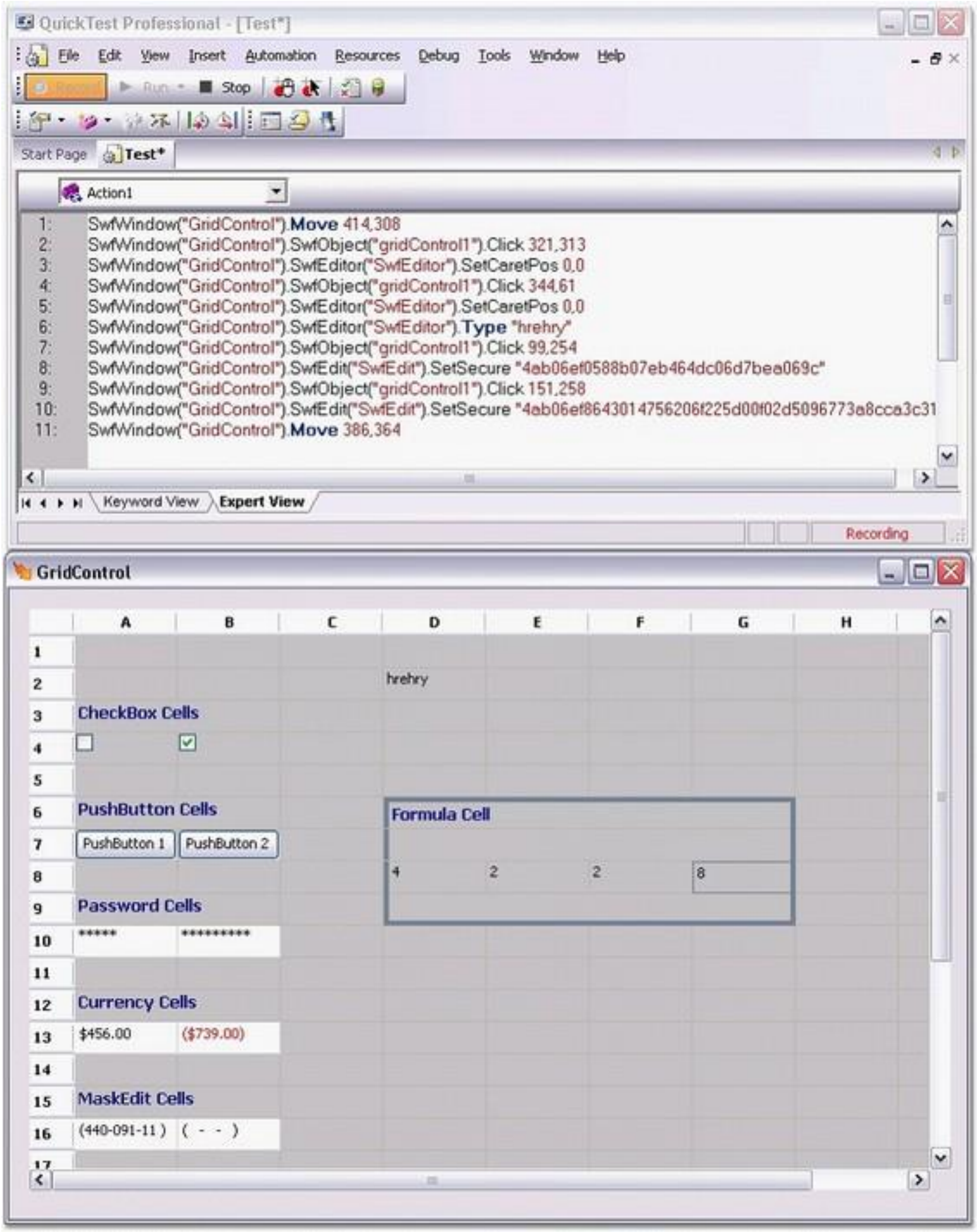
***The DLL renders the correct method names of the Syncfusion namespace that will be called respective to the user-actions performed.***

***These method names are then recorded and displayed in the screen behind the running application, as shown below.***



```
SwfWindow("GridControl").Activate
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 14,4
SwfWindow("GridControl").Move 336,246
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 15,6
```

***This is called high-level recording, as the events are recorded with the method names of the Syncfusion namespace after recognizing the Syncfusion control, unlike the low-level recording in which the Syncfusion controls are not recognized by QTP and the events are recorded with default method names as shown below.***



```
SwfWindow("GridControl").Move 414,308
SwfWindow("GridControl").SwfObject("gridControl1").Click 321,313
SwfWindow("GridControl").SwfEditor("SwfEditor").SetCaretPos 0,0
SwfWindow("GridControl").SwfObject("gridControl1").Click 344,61
SwfWindow("GridControl").SwfEditor("SwfEditor").SetCaretPos 0,0
SwfWindow("GridControl").SwfEditor("SwfEditor").Type "hrehry"
SwfWindow("GridControl").SwfObject("gridControl1").Click 99,254
SwfWindow("GridControl").SwfEdit("SwfEdit").SetSecure "4ab06ef0588b07eb464dc06d7bea069c"
SwfWindow("GridControl").SwfObject("gridControl1").Click 151,258
SwfWindow("GridControl").SwfEdit("SwfEdit").SetSecure "4ab06ef8643014756206f225d00f02d5096773a8cca3c31
SwfWindow("GridControl").Move 386,364
```

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | |
| 2 | | | | hrehry | | | | |
| 3 | CheckBox Cells | | | | | | | |
| 4 | □ | ☑ | | | | | | |
| 5 | | | | | | | | |
| 6 | PushButton Cells | | | Formula Cell | | | | |
| 7 | PushButton 1 | PushButton 2 | | | | | | |
| 8 | | | | 4 | 2 | 2 | 8 | |
| 9 | Password Cells | | | | | | | |
| 10 | ***** | ********* | | | | | | |
| 11 | | | | | | | | |
| 12 | Currency Cells | | | | | | | |
| 13 | $456.00 | ($739.00) | | | | | | |
| 14 | | | | | | | | |
| 15 | MaskEdit Cells | | | | | | | |
| 16 | (440-091-11 ) | ( - - ) | | | | | | |
| 17 | | | | | | | | |

*Figure 18: Default recording without recognizing Syncfusion control*

***The low-level recording is the default recording, which is done by QTP when the steps mentioned in the Configuring Essential QuickTest Professional section are not followed. The recording can be stopped by clicking the Stop button in the toolbar.***

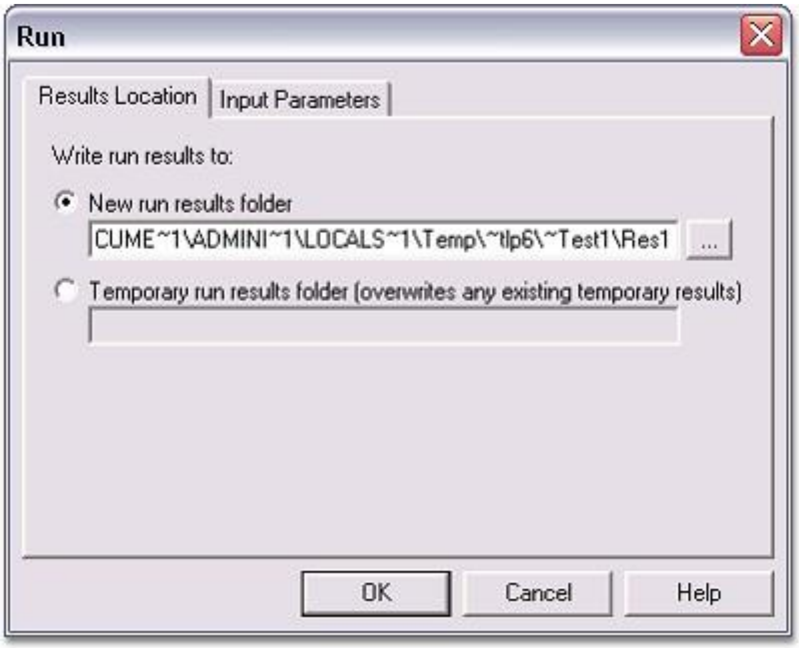
The process of creating and recording the test is completed.

## 3.2 Running a Test

On recording, all the user actions performed in the control are just noted with the corresponding method names of the Syncfusion namespace. The errors can be checked while running a test. To run a test, follow the steps below:

1. Click **Run** in the toolbar.

***Note: The Run dialog box is displayed. The Results Location tab is highlighted by default.***



Run

Results Location | Input Parameters

Write run results to:

● New run results folder

CUME~1\ADMINI~1\LOCALS~1\Temp\~tlp6\~Test1\Res1

○ Temporary run results folder (overwrites any existing temporary results)

*Figure 19: Run Dialog*

In the **Results Location** tab, two options are provided:

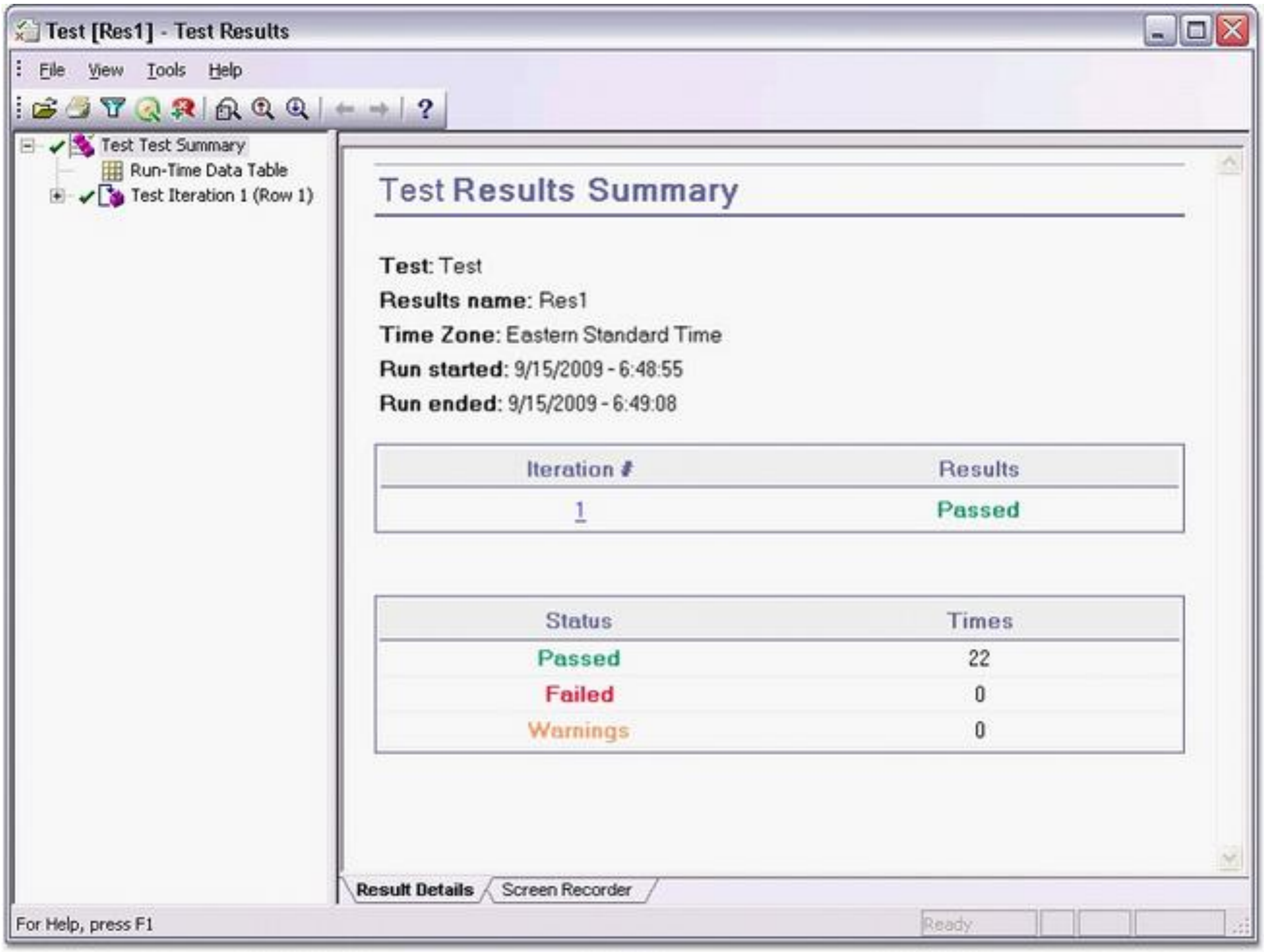
- New run results folder: Allows the results of the test to be written to the location mentioned in the text box below it.
- Temporary run results folder (overwrites any existing temporary results) - Allows the results to be stored in the temporary location.

2. Click the required option.

***Note: Selecting one option renders the other unavailable.***

3. Browse and select the required location by clicking the <icon>

***Note: QTP starts the running process; the application containing the recorded Syncfusion control is opened and it shows all the recorded events in a continuous flow one by one. After it finishes running the test, it displays the Test [Result_Written_Location] - Test Results dialog box, in which the results are summarized as shown below:***



Test [Res1] - Test Results

Test Results Summary

**Test:** Test
**Results name:** Res1
**Time Zone:** Eastern Standard Time
**Run started:** 9/15/2009 - 6:48:55
**Run ended:** 9/15/2009 - 6:49:08

| Iteration # | Results |
|---|---|
| 1 | Passed |

| Status | Times |
|---|---|
| Passed | 22 |
| Failed | 0 |
| Warnings | 0 |

*Figure 20: Test Results*

The process of running the test is completed.

To know more about running scripts, refer to the QTP help document.

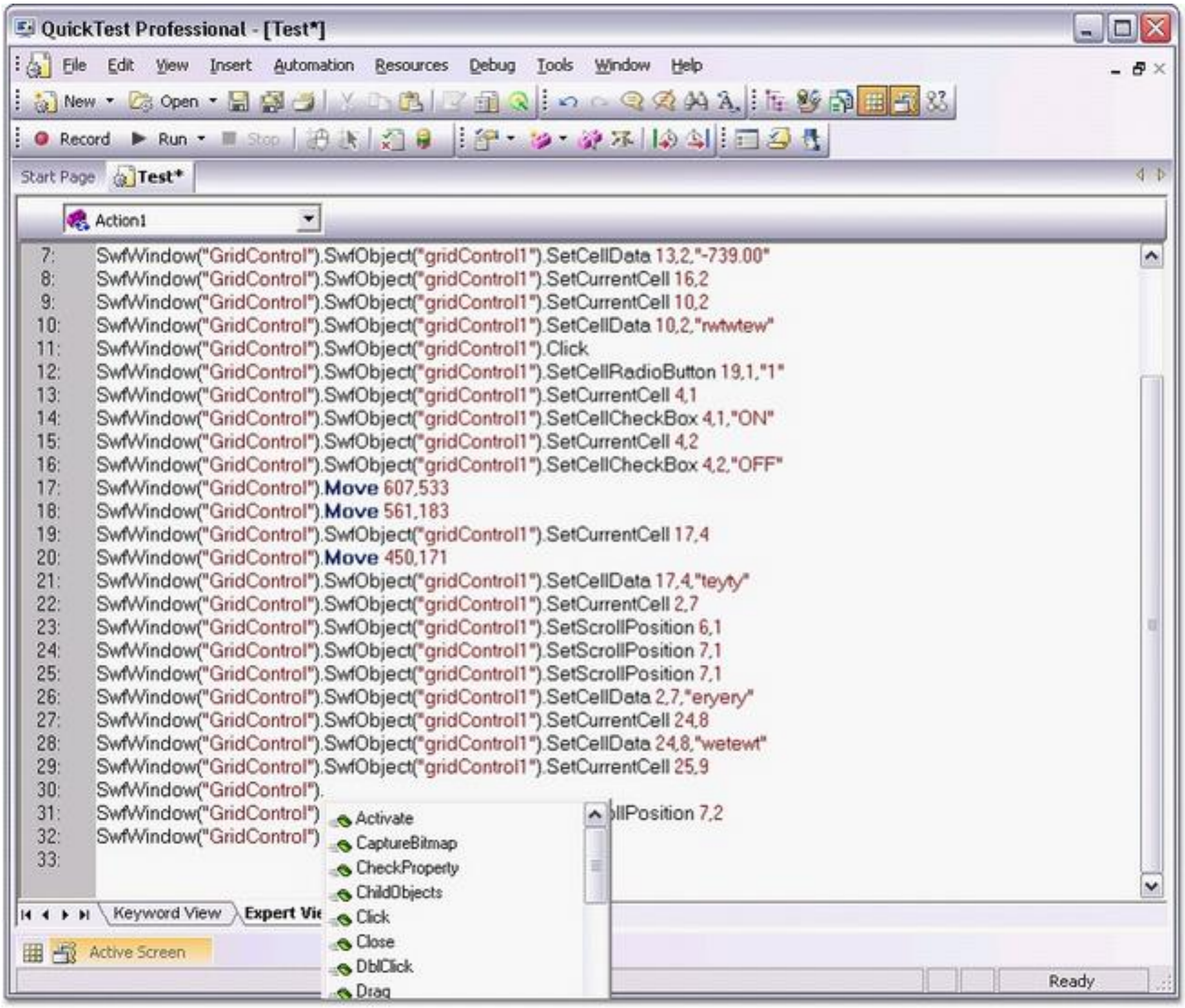
## 3.3 Editing a Test

A test can be edited in either the Keyword view or in the Expert view. You can switch between these views by selecting the required tab at the bottom left of the QuickTest Professional test screen.

### Editing in Expert View

This view is specially provided for the experts in VB script. In the Expert view, the VB scripts are generated while recording. You can also manually write scripts to the existing scripts in this view. So, this view can be used as a tool for managing the testing process in a more controlled manner. You can add scripts to trigger events manually.

The following image shows adding a script line to the Expert View pane.

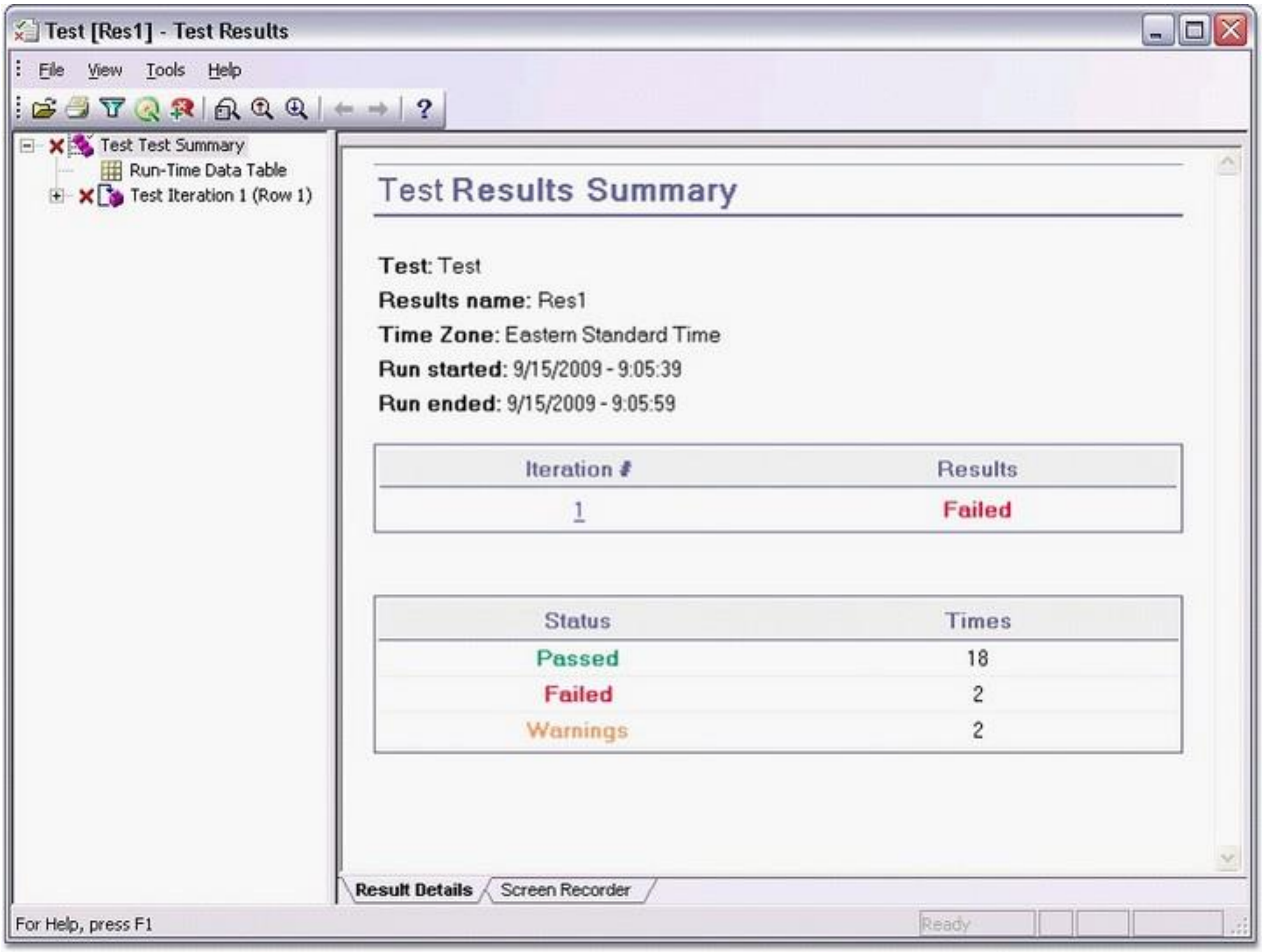


```
SwfWindow("GridControl").SwfObject("gridControl1").SetCellData 13,2,"-739.00"
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 16,2
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 10,2
SwfWindow("GridControl").SwfObject("gridControl1").SetCellData 10,2,"rwtwtew"
SwfWindow("GridControl").SwfObject("gridControl1").Click
SwfWindow("GridControl").SwfObject("gridControl1").SetCellRadioButton 19,1,"1"
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 4,1
SwfWindow("GridControl").SwfObject("gridControl1").SetCellCheckBox 4,1,"ON"
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 4,2
SwfWindow("GridControl").SwfObject("gridControl1").SetCellCheckBox 4,2,"OFF"
SwfWindow("GridControl").Move 607,533
SwfWindow("GridControl").Move 561,183
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 17,4
SwfWindow("GridControl").Move 450,171
SwfWindow("GridControl").SwfObject("gridControl1").SetCellData 17,4,"teyty"
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 2,7
SwfWindow("GridControl").SwfObject("gridControl1").SetScrollPosition 6,1
SwfWindow("GridControl").SwfObject("gridControl1").SetScrollPosition 7,1
SwfWindow("GridControl").SwfObject("gridControl1").SetScrollPosition 7,1
SwfWindow("GridControl").SwfObject("gridControl1").SetCellData 2,7,"eryery"
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 24,8
SwfWindow("GridControl").SwfObject("gridControl1").SetCellData 24,8,"wetewt"
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 25,9
SwfWindow("GridControl").
SwfWindow("GridControl")                          llPosition 7,2
SwfWindow("GridControl")
```

Figure 21: Editing in Expert view

You can run the edited test to check whether the newly added or changed scripts affect the running process by showing the changes in the running application.

***Note: Sometimes, the newly added or changed script may have an error causing the whole application to fail. In such a case, the Test Results dialog will show the failure as shown below:***



Test Results Summary

**Test:** Test
**Results name:** Res1
**Time Zone:** Eastern Standard Time
**Run started:** 9/15/2009 - 9:05:39
**Run ended:** 9/15/2009 - 9:05:59

| Iteration # | Results |
|---|---|
| 1 | Failed |

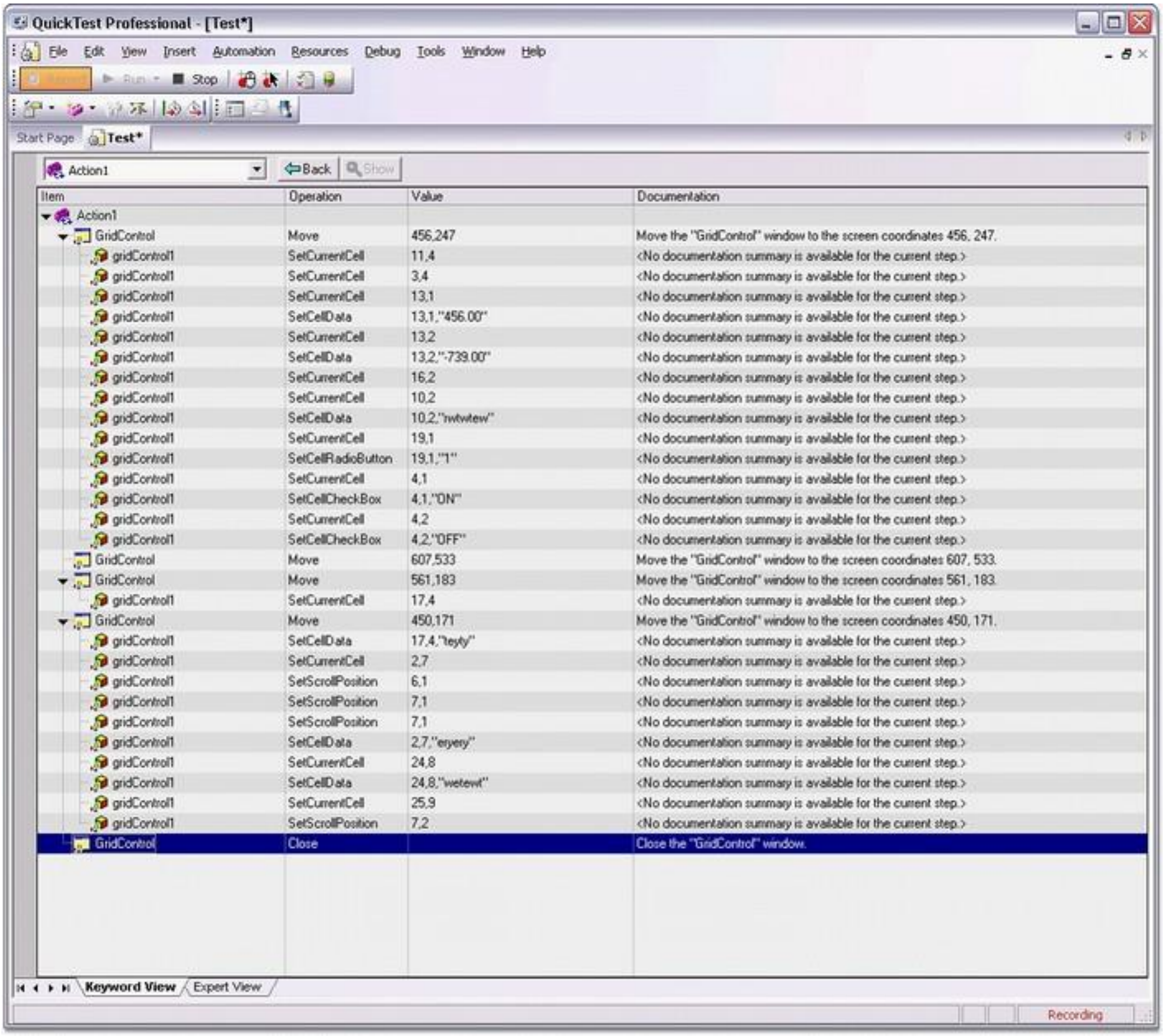
| Status | Times |
|---|---|
| Passed | 18 |
| Failed | 2 |
| Warnings | 2 |

Figure 22: Test Results when Testing Fails

For more details on running the test, refer to the previous section.

### Editing in Keyword View

The keyword view is meant for persons who are not experts in VB scripts. Keyword view contains the controls used, the user-actions or operations performed, values involved in the operation, and the documentation summary in a table format. The controls used are listed under the Item header in a tree-view format as shown below.



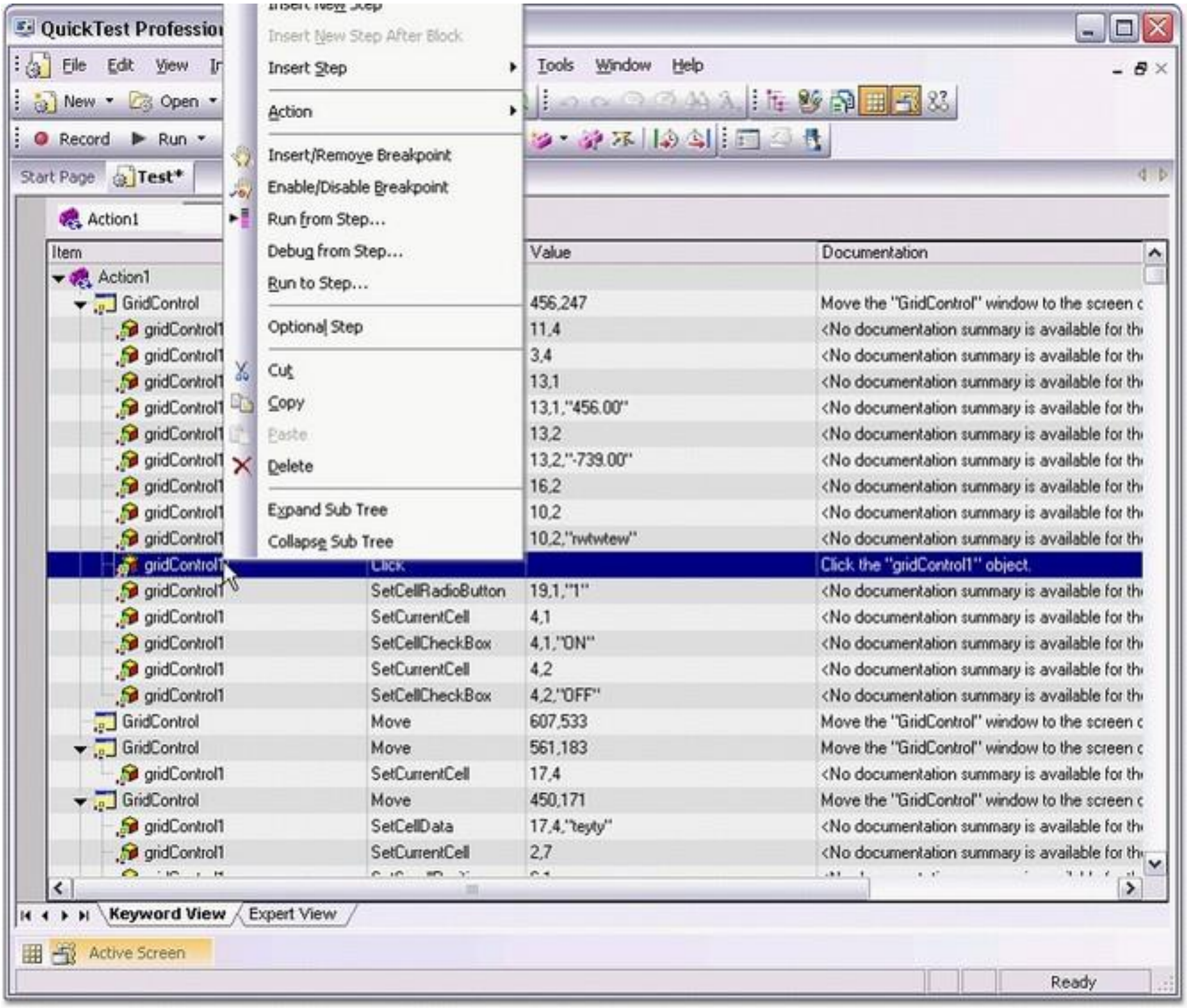
QuickTest Professional - [Test*]
File Edit View Insert Automation Resources Debug Tools Window Help
Record Run Stop
Start Page Test*
Action1
Back Show
Keyword View Expert View
Recording

| Item | Operation | Value | Documentation |
|---|---|---|---|
| Action1 | | | |
| GridControl | Move | 456,247 | Move the "GridControl" window to the screen coordinates 456, 247. |
| gridControl1 | SetCurrentCell | 11,4 | <No documentation summary is available for the current step.> |
| gridControl1 | SetCurrentCell | 3,4 | <No documentation summary is available for the current step.> |
| gridControl1 | SetCurrentCell | 13,1 | <No documentation summary is available for the current step.> |
| gridControl1 | SetCellData | 13,1,"456.00" | <No documentation summary is available for the current step.> |
| gridControl1 | SetCurrentCell | 13,2 | <No documentation summary is available for the current step.> |
| gridControl1 | SetCellData | 13,2,"-739.00" | <No documentation summary is available for the current step.> |
| gridControl1 | SetCurrentCell | 16,2 | <No documentation summary is available for the current step.> |
| gridControl1 | SetCurrentCell | 10,2 | <No documentation summary is available for the current step.> |
| gridControl1 | SetCellData | 10,2,"rwtwtew" | <No documentation summary is available for the current step.> |
| gridControl1 | SetCurrentCell | 19,1 | <No documentation summary is available for the current step.> |
| gridControl1 | SetCellRadioButton | 19,1,"1" | <No documentation summary is available for the current step.> |
| gridControl1 | SetCurrentCell | 4,1 | <No documentation summary is available for the current step.> |
| gridControl1 | SetCellCheckBox | 4,1,"ON" | <No documentation summary is available for the current step.> |
| gridControl1 | SetCurrentCell | 4,2 | <No documentation summary is available for the current step.> |
| gridControl1 | SetCellCheckBox | 4,2,"OFF" | <No documentation summary is available for the current step.> |
| GridControl | Move | 607,533 | Move the "GridControl" window to the screen coordinates 607, 533. |
| GridControl | Move | 561,183 | Move the "GridControl" window to the screen coordinates 561, 183. |
| gridControl1 | SetCurrentCell | 17,4 | <No documentation summary is available for the current step.> |
| GridControl | Move | 450,171 | Move the "GridControl" window to the screen coordinates 450, 171. |
| gridControl1 | SetCellData | 17,4,"teyty" | <No documentation summary is available for the current step.> |
| gridControl1 | SetCurrentCell | 2,7 | <No documentation summary is available for the current step.> |
| gridControl1 | SetScrollPosition | 6,1 | <No documentation summary is available for the current step.> |
| gridControl1 | SetScrollPosition | 7,1 | <No documentation summary is available for the current step.> |
| gridControl1 | SetScrollPosition | 7,1 | <No documentation summary is available for the current step.> |
| gridControl1 | SetCellData | 2,7,"eryery" | <No documentation summary is available for the current step.> |
| gridControl1 | SetCurrentCell | 24,8 | <No documentation summary is available for the current step.> |
| gridControl1 | SetCellData | 24,8,"wetewt" | <No documentation summary is available for the current step.> |
| gridControl1 | SetCurrentCell | 25,9 | <No documentation summary is available for the current step.> |
| gridControl1 | SetScrollPosition | 7,2 | <No documentation summary is available for the current step.> |
| GridControl | Close | | Close the "GridControl" window. |

*Figure 23: Keyword View*

To edit the test in Keyword view, you can perform any of the following actions:

1. You can right-click any of the items listed under the Item header and choose one of the options available in the displayed menu as shown below.

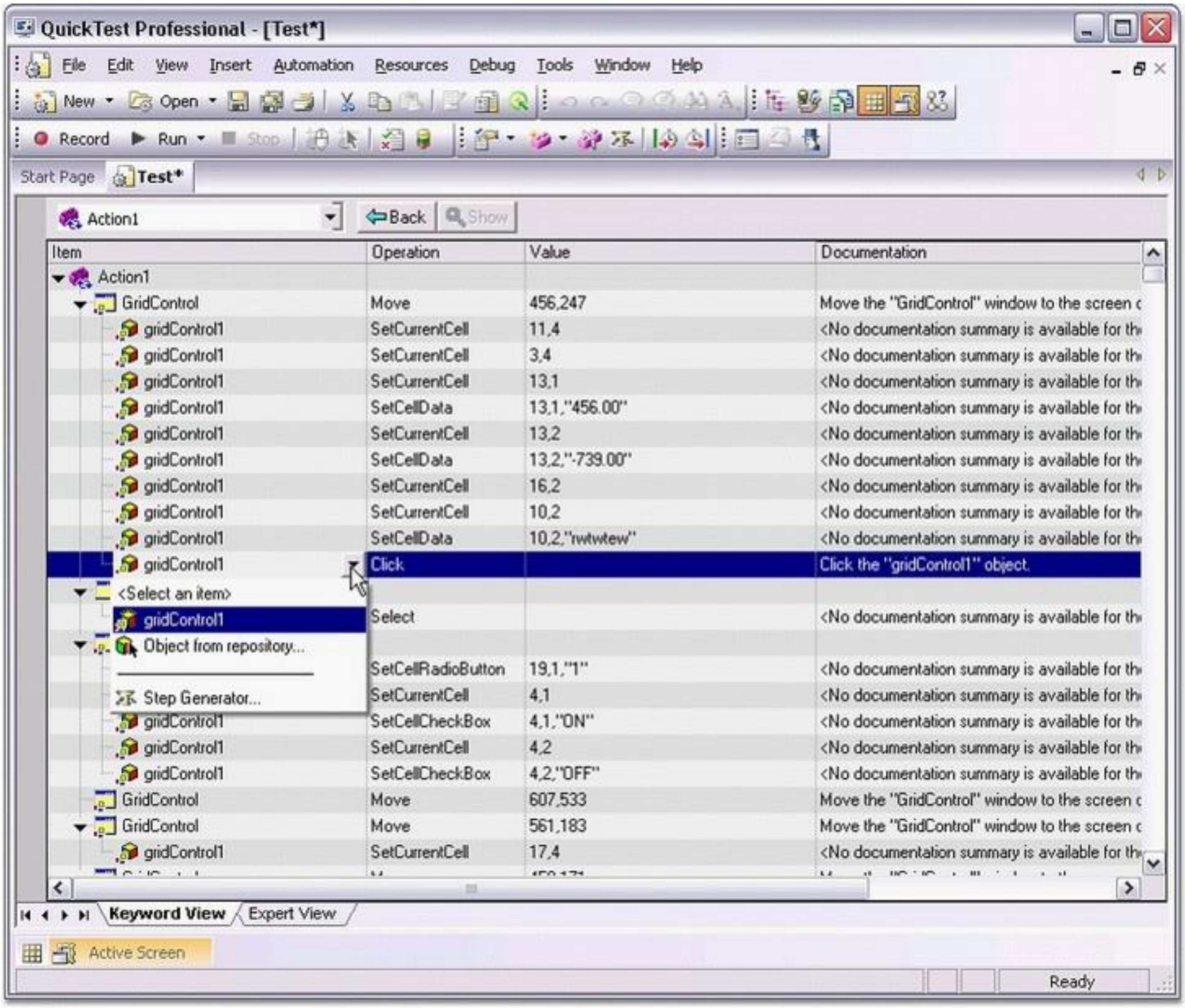


| Item | | Value | Documentation |
|---|---|---|---|
| Action1 | | | |
| GridControl | | 456,247 | Move the "GridControl" window to the screen c |
| gridControl1 | | 11,4 | <No documentation summary is available for th |
| gridControl1 | | 3,4 | <No documentation summary is available for th |
| gridControl1 | | 13,1 | <No documentation summary is available for th |
| gridControl1 | | 13,1,"456.00" | <No documentation summary is available for th |
| gridControl1 | | 13,2 | <No documentation summary is available for th |
| gridControl1 | | 13,2,"-739.00" | <No documentation summary is available for th |
| gridControl1 | | 16,2 | <No documentation summary is available for th |
| gridControl1 | | 10,2 | <No documentation summary is available for th |
| gridControl1 | | 10,2,"rwtwtew" | <No documentation summary is available for th |
| gridControl1 | Click | | Click the "gridControl1" object. |
| gridControl1 | SetCellRadioButton | 19,1,"1" | <No documentation summary is available for th |
| gridControl1 | SetCurrentCell | 4,1 | <No documentation summary is available for th |
| gridControl1 | SetCellCheckBox | 4,1,"ON" | <No documentation summary is available for th |
| gridControl1 | SetCurrentCell | 4,2 | <No documentation summary is available for th |
| gridControl1 | SetCellCheckBox | 4,2,"OFF" | <No documentation summary is available for th |
| GridControl | Move | 607,533 | Move the "GridControl" window to the screen c |
| GridControl | Move | 561,183 | Move the "GridControl" window to the screen c |
| gridControl1 | SetCurrentCell | 17,4 | <No documentation summary is available for th |
| GridControl | Move | 450,171 | Move the "GridControl" window to the screen c |
| gridControl1 | SetCellData | 17,4,"teyty" | <No documentation summary is available for th |
| gridControl1 | SetCurrentCell | 2,7 | <No documentation summary is available for th |

*Figure 24: Editing in Keyword View – Right-click*

For example, clicking **Cut** in the menu will cause the row representing a user-action to be cut. You can then right-click on any other item and click **Paste** on the menu displayed. This causes the row to be pasted before the right-clicked item.

***Note: All the items under the Item header are represented as a drop-down list.***



| Item | Operation | Value | Documentation |
|---|---|---|---|
| Action1 | | | |
| GridControl | Move | 456,247 | Move the "GridControl" window to the screen c |
| gridControl1 | SetCurrentCell | 11,4 | <No documentation summary is available for th |
| gridControl1 | SetCurrentCell | 3,4 | <No documentation summary is available for th |
| gridControl1 | SetCurrentCell | 13,1 | <No documentation summary is available for th |
| gridControl1 | SetCellData | 13,1,"456.00" | <No documentation summary is available for th |
| gridControl1 | SetCurrentCell | 13,2 | <No documentation summary is available for th |
| gridControl1 | SetCellData | 13,2,"-739.00" | <No documentation summary is available for th |
| gridControl1 | SetCurrentCell | 16,2 | <No documentation summary is available for th |
| gridControl1 | SetCurrentCell | 10,2 | <No documentation summary is available for th |
| gridControl1 | SetCellData | 10,2,"rwtwtew" | <No documentation summary is available for th |
| gridControl1 | Click | | Click the "gridControl1" object. |
| <Select an item> | | | |
| | Select | | <No documentation summary is available for th |
| | | | |
| | SetCellRadioButton | 19,1,"1" | <No documentation summary is available for th |
| | SetCurrentCell | 4,1 | <No documentation summary is available for th |
| gridControl1 | SetCellCheckBox | 4,1,"ON" | <No documentation summary is available for th |
| gridControl1 | SetCurrentCell | 4,2 | <No documentation summary is available for th |
| gridControl1 | SetCellCheckBox | 4,2,"OFF" | <No documentation summary is available for th |
| GridControl | Move | 607,533 | Move the "GridControl" window to the screen c |
| GridControl | Move | 561,183 | Move the "GridControl" window to the screen c |
| gridControl1 | SetCurrentCell | 17,4 | <No documentation summary is available for th |

*Figure 25: Editing on Keyword View – Drop-down*

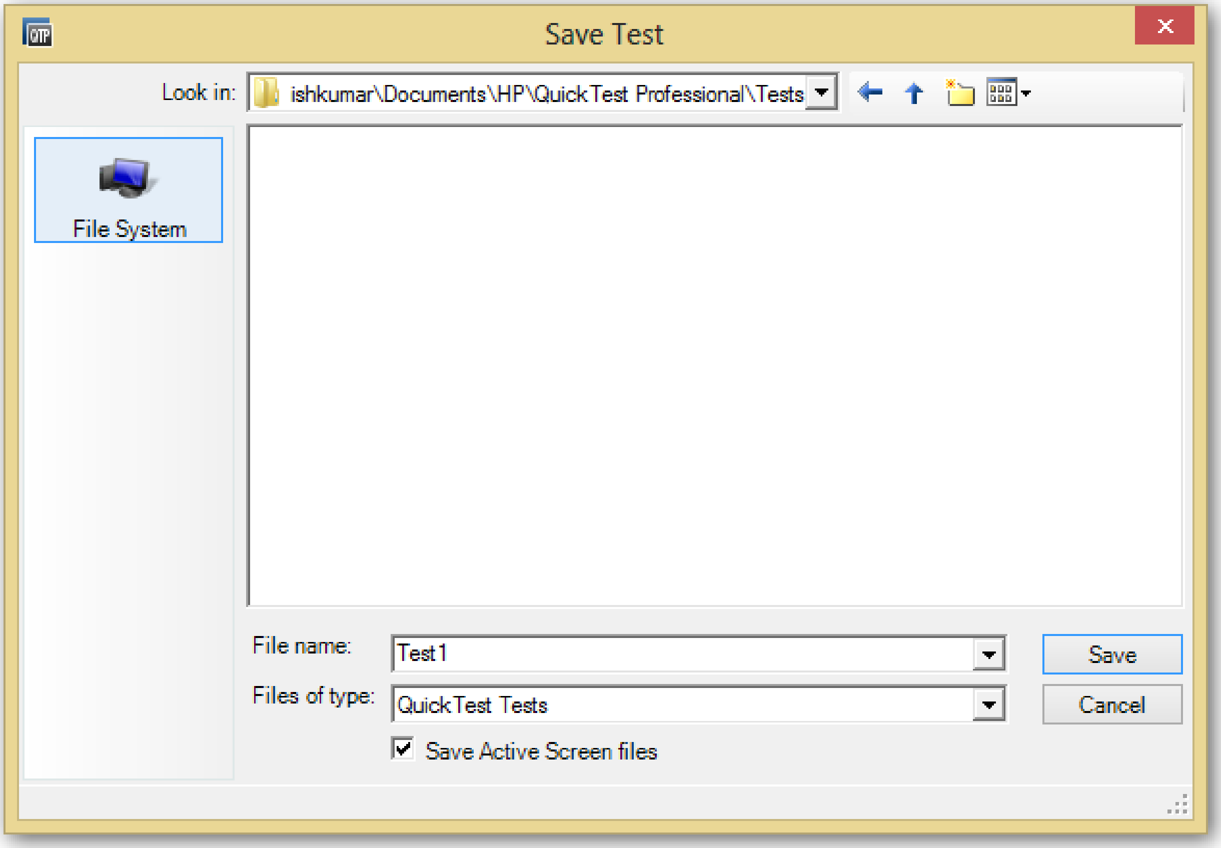
You can then run the edited test.

For more details on running the edited test, refer to **Editing on Expert View** topic.

## 3.4 Saving a Test

Saving a test is like saving any other document or picture. To save a test, follow the steps below:

1. Click the **Save** button in the toolbar. The **Save Test** dialog box is displayed.



ishkumar\Documents\HP\QuickTest Professional\Tests

File name: Test1

Files of type: QuickTest Tests

☑ Save Active Screen files

*Figure 26: Save Test Dialog*

2. Select the location to save the file from the **Save in:** drop-down list.
3. Type the file name of the file to be saved in the text box adjacent to the **File name** label.
4. Click **Save**.

The test is saved.

## 3.5 Running the Saved Test

The tests that have been saved can be replayed later. To run a saved test, follow the steps below:

1. Click **Open** on the toolbar.

***Note: The Open Test dialog box is displayed with a list of saved tests.***

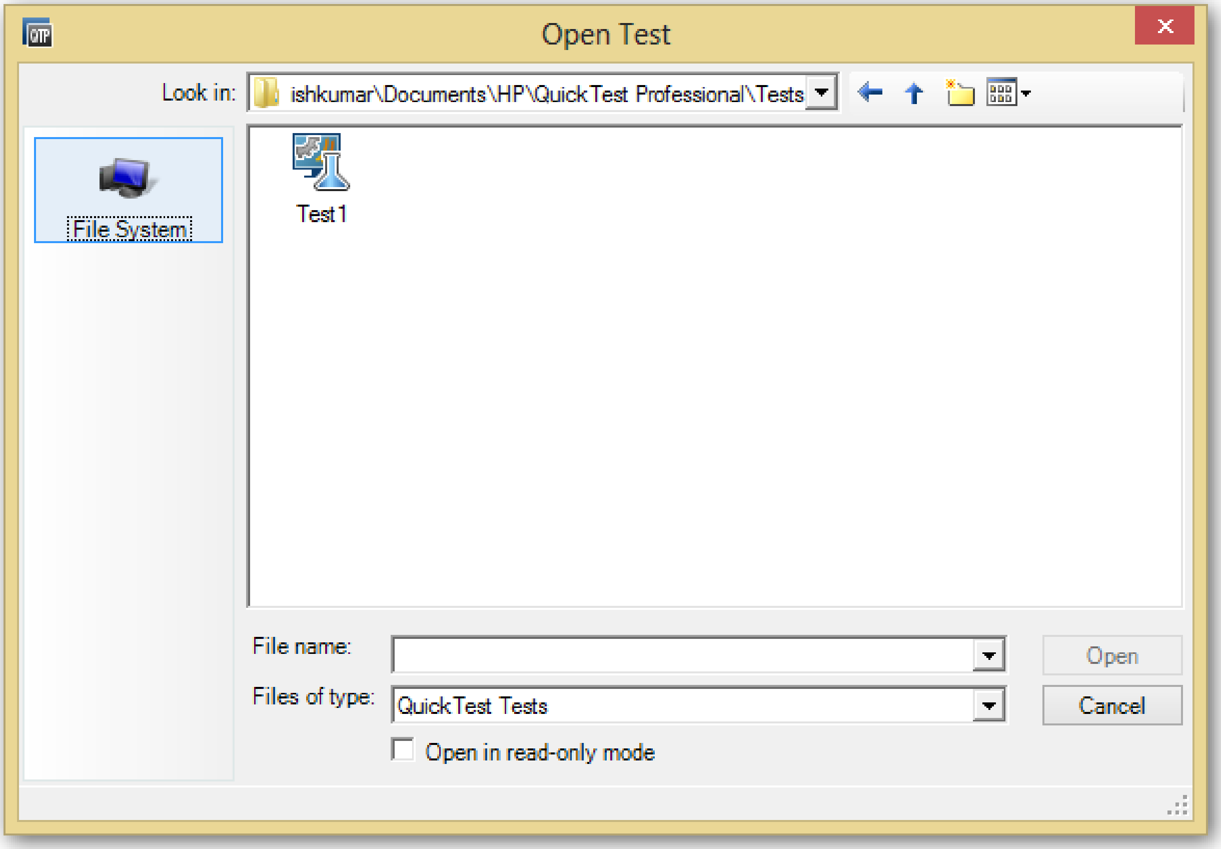
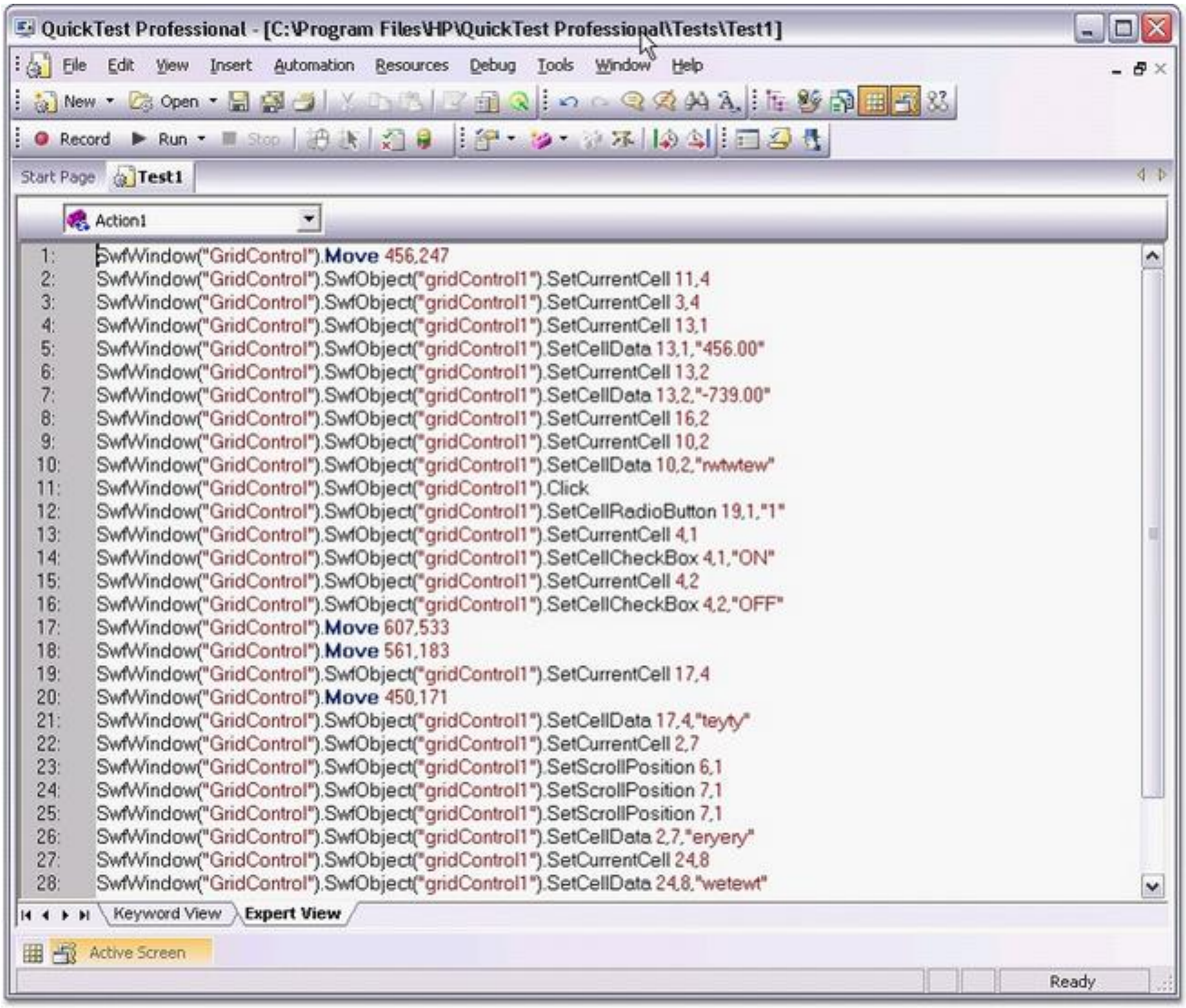


*Figure 27: Open Test Dialog*

2. Select the required test and click **Open**.

***Note: The saved test is opened with its name and the complete path as the name of the window. By default, Expert View of the Test is opened.***

The following image shows the mouse pointer pointing towards the path and file displayed as the window name.



QuickTest Professional - [C:\Program Files\HP\QuickTest Professional\Tests\Test1]
File Edit View Insert Automation Resources Debug Tools Window Help
New Open
Record Run Stop
Start Page Test1
Action1
Keyword View Expert View
Active Screen
Ready

```
SwfWindow("GridControl").Move 456,247
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 11,4
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 3,4
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 13,1
SwfWindow("GridControl").SwfObject("gridControl1").SetCellData 13,1,"456.00"
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 13,2
SwfWindow("GridControl").SwfObject("gridControl1").SetCellData 13,2,"-739.00"
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 16,2
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 10,2
SwfWindow("GridControl").SwfObject("gridControl1").SetCellData 10,2,"rwtwtew"
SwfWindow("GridControl").SwfObject("gridControl1").Click
SwfWindow("GridControl").SwfObject("gridControl1").SetCellRadioButton 19,1,"1"
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 4,1
SwfWindow("GridControl").SwfObject("gridControl1").SetCellCheckBox 4,1,"ON"
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 4,2
SwfWindow("GridControl").SwfObject("gridControl1").SetCellCheckBox 4,2,"OFF"
SwfWindow("GridControl").Move 607,533
SwfWindow("GridControl").Move 561,183
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 17,4
SwfWindow("GridControl").Move 450,171
SwfWindow("GridControl").SwfObject("gridControl1").SetCellData 17,4,"teyty"
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 2,7
SwfWindow("GridControl").SwfObject("gridControl1").SetScrollPosition 6,1
SwfWindow("GridControl").SwfObject("gridControl1").SetScrollPosition 7,1
SwfWindow("GridControl").SwfObject("gridControl1").SetScrollPosition 7,1
SwfWindow("GridControl").SwfObject("gridControl1").SetCellData 2,7,"eryery"
SwfWindow("GridControl").SwfObject("gridControl1").SetCurrentCell 24,8
SwfWindow("GridControl").SwfObject("gridControl1").SetCellData 24,8,"wetewt"
```

*Figure 28: Test Opened*

3. Click **Run** on the toolbar to run the test.

For more details on running the test, refer to the **Running a Test** topic in this document.

The process of running a saved test is complete.

# 4 Supported Controls and Methods

The following controls are supported by Essential QuickTest Professional:

- Essential Grid
- Essential Tools
- Essential Chart
- Essential Schedule
- Essential Diagram

By supported methods, we mean those methods that are recorded in QTP.

## 4.1 Essential Grid

Essential Grid supports the following controls:

- GridControl
- GridDataBoundGrid
- GridGroupingControl
- GridListControl
- TabBarSplitterControl

The following are the recorded methods and their corresponding descriptions for Essential Grid:

**Grid Control**

| Method | Description |
|---|---|
| CellButtonClick(int row, int col) | Raises the click on the cell button. |
| CellDoubleClick(int row, int col) | Raises the cell double-click. |
| GetDescription(int row, int col) | Gets the description of grid cells. |
| MouseDown(int row, int col, string button) | Raises a click in the grid. |
| MoveColumn(int fromColumn, int count, int target) | Moves a range of columns. |
| MoveRow(int from, int count, int target) | Moves a range of rows from the specified location to a target location. |
| ResizeColumn(int fromColumn, int to, int width) | Resizes the specified columns. |
| ResizeRow(int fromRow, int to, int height) | Resizes the specified rows. |
| SelectRange(string range, int top, int left, int | Selects the range. |

| | |
|---|---|
| bottom, int right) | |
| SetCellData(int row, int col, string str) | Sets the cell value of the cell. |
| SetCellCheckBox(int row, int col, string str) | Sets the cell value of the check box cell. |
| SetCellRadioButton(int row, int col, string str) | Sets the cell value of the radio button cell. |
| SetCurrentCell(int row, int col) | Sets the location of the current cell. |
| SetScrollPosition(int vScrollPosition, int hScrollPosition) | Sets the scroll position. |
| **Helper Functions** | |
| BeginEdit(int row, int col) | Brings the editing cursor in the specified grid cell. |
| EndEdit(int row, int col) | Finishes the editing mode of the cell specified. |
| string GetCellType(int row, int col) | Retrieves the CellType for the given cell coordinates. |
| int GetColumnCount() | Retrieves the number of columns used. |
| int GetColumnIndex(string name) | Finds the column index for the given column name, returns 0 if search fails. |
| string GetFormattedText(int row, int col) | Retrives the formatted cell format. |
| bool IsFormulaCell(int row, int col, out string formula, out string computedValue) | For a given row and column index, IsFormulaCell points to the formula used in that cell and the result of the formula. This also returns “false” if this cell is not a formula cell. |
| object GetCellData(int row, int col) | For the given Row and Column objects, the cell value of that cell can be obtained. |
| int GetRowCount() | Retrieves the number of rows used. |
| InsertColumn(int insertAt, int count) | Inserts a range of columns from the specified location. |
| InsertRow(int insertAt, int count) | Inserts a range of rows from the specified location. |
| RemoveColumn(int from, int to) | Removes a range of columns specified for the Grid |

| | |
|---|---|
| | control. |
| RemoveRow(int from, int to) | Removes a range of rows specified for the Grid control. |
| ScrollToCell(int rowIndex, int colIndex) | Scrolls the grid so that the cell will be visible for replay. |
| HideRow(int from, int to) | Hides a range of rows specified for the Grid control. |
| ShowHiddenRow(int from, int to) | Shows a range of rows specified for the Grid control, which were hiding. |
| HideCol(int from, int to) | Hides a range of columns specified for the Grid control. |
| ShowHiddenCol(int from, int to) | Shows a range of columns specified for the Grid control. |
| int GetSelectedRowIndex() | Returns the top row index of the selected row. |
| GetSelectedColIndex() | Returns the column index of the selected column. |
| Color GetCellBackColor(int row, int col) | Gets the back color of the cell. |
| string GetName() | Gets the name of the Grid control object. |

**GridDataBoundGrid**

| Method | Description |
|---|---|
| CellButtonClick(int row, int col) | Raises the click on the cell button. |
| CellDoubleClick(int row, int col) | Raises the cell double-click. |
| CollapseRow(int rowIndex) | Collapses the row for the specified row index. |
| DeleteRow(int from, int to) | Deletes the specified rows. |
| ExpandRow(int rowIndex) | Expands the Row for the specified row index. |
| MouseDown(int row, int col, string button) | Raises a click in the grid. |
| MoveColumn(int fromColumn, int count, int target) | Moves a range of columns. |

| | |
|---|---|
| ResizeColumn(int fromColumn, int to, int width) | Resizes the specified columns. |
| ResizeRow(int fromRow, int to, int height) | Resizes the specified rows. |
| SelectRange(string range, int top, int left, int bottom, int right) | Selects the range. |
| SetCellData(int row, int col, string str) | Sets the cell value of the cell. |
| SetCellCheckBox(int row, int col, string str) | Sets the cell value of the check box cell. |
| SetCellRadioButton(int row, int col, string str) | Sets the cell value of the radio button cell. |
| SetCurrentCell(int row, int col) | Sets the location of current cell. |
| SetScrollPosition(int vScrollPosition, int hScrollPosition) | Sets the scroll position. |
| SortColumn(int col, string sortBehavior) | Sorts the column. |
| **Helper Functions** | |
| BeginEdit(int row, int col) | Brings the editing cursor in the specified grid cell. |
| string GetCellType(int row, int col) | Retrieves the CellType for the given cell co-ordinates. |
| string GetCellBackColor(int row, int col) | Retrieves the Back color for the given cell co-ordinates. |
| int GetColumnCount() | Retrieves the number of columns used. |
| int GetVisibleColumnCount() | Retrieves the number of visible columns. |
| int GetColumnIndex(string name) | Finds the column index for the given column name, returns 0 if search fails. |
| Int GetCurrentCellImageIndex(int row, int col) | Gets the image index of the current cell. |
| string GetFormattedText(int row, int col) | Retrieves the formatted cell value. |
| bool IsColumnVisible(int col) | Checks if the column is visible. |
| bool IsFormulaCell(int row, int col, out string formula, out string computedValue) | For a given row and column index, IsFormulaCell points to the formula used in that cell and the result |

| | |
|---|---|
| | of the formula. This also returns “false” if this cell is not a formula cell. |
| object GetCellData(int row, object col) | For the given Row and Column objects, the cell value of that cell can be obtained. |
| int GetRowCount() | Retrieves the number of rows used. |
| ScrollToCell(int rowIndex, int colIndex) | Scrolls the grid so as the cell to be visible for replay. |
| HideRow(int from, int to) | Hides a range of rows specified for the GridControl. |
| HideCol(int from, int to) | Hides a range of columns specified for the GridControl. |
| int GetSelectedRowIndex() | Returns the top row index of the selected row. |
| int GetSelectedColIndex() | Returns the column index of the selected column. |
| int GetSelectedRowCount() | Returns the number of selected rows. |
| int GetSelectedColCount() | Returns the number of selected columns. |
| string GetSelectedRowRange() | Returns the Top and Bottom row of the selected row range. |
| string GetSelectedColRange() | Returns the left and right column of the selected column range. |
| bool IsColSorted(int col) | Determines whether the column is sorted. |
| string GetColSortOrder(int col) | Returns the sort order of the sorted column (Ascending or Descending). |
| string GetName() | Gets the name of the Grid DataBoundGrid object |

**GridGroupingControl**

| Method | Description |
|---|---|
| CellButtonClick(object row, object col) | Raises the cell button click. |
| CellDoubleClick(object row,object col) | Raises the cell double-click. |
| CollapseRecord(object record) | Collapses the record. |

| | |
|---|---|
| CollapseGroup(object row) | Collapses the group. |
| ExpandGroup(object row) | Expands the group. |
| ExpandRecord(object record) | Expands the record. |
| FindRecordInGrid(string tableObject, string columnName, string data) | Returns the first index of the searched data for the given column of the table, as located in the NestedDisplayElements. |
| FindRecordInTable(string tableObject, string columnName, string data) | Returns the first index of the searched data for the given column. |
| GetAbsoluteRowIndex(int RowIndex) | Retrieves the absolute RowIndex. |
| GetBackColor(int row) | Gets the backcolor of the record. |
| GetCellBackColor(object row, object col) | Gets the backcolor of the Cell. |
| GetCellData(object row, object col) | For the given Row and Column objects, the cell value of that cell can be obtained. |
| GetChildCount(object row) | Gets the child count for the given caption row and a record row. |
| GetDescription(object row, object col) | Gets the description of grid cells. |
| GetColumnCount() | Returns the sort order of the sorted column (Ascending or Descending). |
| GetColSortOrder(int col) | Returns the sort order of the sorted column (Ascending or Descending). |
| GetColumnName(string tablename, int colindex) | For a given table name and column index, the column name in which an element resides can be obtained. |
| GetDetails() | Gets details like table, record, and table descriptor. |
| GetLevelByTableName(string name) | Gets the level of table for the given table name. |
| GetRowCount() | Retrieves the number of rows used. |
| GetRowElement(object row) | Gets the row element. |
| GetSelectedColIndex() | Returns the Left column index of the selected |

| | |
|---|---|
| | columns. |
| GetSelectedRowIndex() | Returns the top row index of the selected row. |
| GetSelectedRowRange() | Returns the Top and Bottom row of the selected row range. |
| GetTableName(object row) | Obtains the table name for a given Row. |
| GetTableNameByLevel(int level) | Gets the level of the table for the given table name. |
| GroupBy(string tablename,string column, string status) | Defines grouping and ungrouping of specified columns. |
| MouseDown(object row, object col, string button) | Raises the MouseDown. |
| MouseDownOnRowHeader(int row, int col, string button) | Raises the MouseDown on the RowHeader. |
| MoveColumn(string tablename, object fromColumn, object count, object target) | Moves a range of columns. |
| IsColSorted(int col) | Determines whether the column is sorted. |
| IsGroupExpanded(object row) | Determines whether the specified group is expanded. |
| IsGroupRow(object row) | Determines whether the specified row is a caption row or caption section. |
| IsRecord(object record) | Determines whether the specified row is a record. |
| IsRecordExpanded(object record) | Determines whether the specified record is expanded. |
| ResizeColumn(string tablename, int fromColumn, int to, int width) | Resizes the specified column. |
| ResizeRow(string tablename, int fromRow, int to, int height) | Resizes the specified rows. |
| SelectRange(string range, int top, int left, int bottom, int right) | Selects the range. |
| SelectRecord(object row, string status) | Selects a record for the GridGroupingControl. |

| | |
|---|---|
| SetCellData(object row, object col, string str) | Sets the cell value of the cell. |
| SetCellCheckBox(object row, object col, string str) | Sets the cell value of the check box cell. |
| SetCellRadioButton(object row, object col, string str) | Sets the cell value of the radio button cell. |
| SetCurrentCell(object row, object col) | Sets the location of current cell. |
| SetScrollPosition(int vScrollPosition, int hScrollPosition) | Sets the scroll position. |
| SortColumn(string tablename,object col, string sortBehavior, bool cntrl) | Sorts the column. |
| SelectRecords(object row, object count) | Selects mutiple records for the GridGroupingControl. |
| ScrollToColumn(string tablename, object col) | The grid will scroll to the given column. |
| ScrollToRow(int row) | The grid will scroll to the given row. |
| AddNewRow(string objn) | A new row will be added. |
| string GetFormattedText(int row, int col) | Retrieves the formatted cell value. |
| string GetName() | Gets the name of the Grid control object. |

**GridListControl**

| Method | Description |
|---|---|
| ResizeColumn(object fromColumn, int to, int width) | Resizes the specified columns. |
| ResizeRow(int fromRow, int to, int height) | Resizes the specified rows. |
| SelectRow(int top,int bottom) | Selects the range. |
| string GetName() | Gets the name of the GridListControl object |

**TabBarSplitterControl**

| Method | Description |
|---|---|
| GetName() | Gets the name of the TabBarControl. |

| | |
|---|---|
| GetTabName(int index) | The label in the tab page can be known by passing the index. |
| Select(string tab) | The name of the selected tab. |
| SetSplitterPosition(string tab, int vSplit, int hSplit) | The splitter position in the tab bar page. |

## 4.2 Essential Tools

The following controls are supported by Essential Tools.

- ButtonAdv
- CalculatorControl
- CheckBoxAdv
- ColorPickerUIAdv
- ComboBoxAutoComplete
- ComboDropDown
- CommandBar
- DataListView
- DateTimePickerAdv
- DockingManager
- GroupBar
- GroupView
- MultiColumnComboBox
- Popupmenu
- ProgressBarAdv
- RadioButtonAdv
- RibbonControlAdv
- ScrollerFrame
- TabbedMDI
- TabControlAdv
- XPTaskBar
- TextBoxExt
- ThemedCheckedButton
- TreeViewAdv
- XPMenus
- XPToolBar
- SplitContainerAdv
- TabSplitterContainer
- TrackBarEx
- RangeSlider

- NavigationView

The following are the recorded methods and their corresponding descriptions for Essential Tools:

### ButtonAdv

| Method | Description |
|---|---|
| Click(string text) | Performs click action on the ButtonAdv control. |

### CalculatorControl

| Method | Description |
|---|---|
| SetValue(int value) | The value will be appended to the calculated value. |
| SetAction(string action) | The action will be paused at the calculated value. |
| double GetCalculatedValue() | Helps to get the current value from the text area. |
| SetCalculatedValue(double value) | Sets the value in the text area as specified in the argument. |

### CheckBoxAdv

| Method | Description |
|---|---|
| Set(string chkState) | The CheckState of the CheckBoxAdv. |
| **Helper Function** | |
| string GetCheckState() | Gets the CheckState of the CheckBoxAdv. |

### ColorPickerUIAdv

| Method | Description |
|---|---|
| SelectColor(object color) | The color that has to be selected. |

### ComboBoxAutoComplete

| Method | Description |
|---|---|
| DropDown() | Shows the drop-down list. |

**ComboDropDown**

| Method | Description |
|---|---|
| DropDown() | Shows the drop-down list. |
| Select(object item) | Selects the item in the list. |

**CommandBar**

| Method | Description |
|---|---|
| DropDown() | Shows the drop-down list. |
| SetDockState(string dockState) | Changes the dock state. |
| SetFloatState(int x, int y) | Sets the CommandBar to float. |

**DataListView**

| Method | Description |
|---|---|
| Select(string item) | Selects the specified item. |
| Return() | Performs click on the focused item. |

**DateTimePickerAdv**

| Method | Description |
|---|---|
| void CheckEnabled(object on, string checkState); | Interface to check the enabled state of the DateTimePickerAdv. |
| void ChangeValue(object on, string dateTime); | Interface to change the value of the DateTimePickerAdv. |

| | |
|---|---|
| void ShowPopupWindow(object visible, object x, object y); | Interface to show the calendar popup. |
| void ShowCalendar(object visible); | Interface to show the calendar in the popup window. |
| void SetCalendarValue(object visible, string calValue); | Interface to set the Calendar value of the DateTimePickerAdv control. |
| void PopupClose(object visible); | Interface to close the popup window. |
| void SetTodayValue(string str); | Interface to set the today value when the today button is clicked. |
| void SetNoValue(string str); | Interface to set the null value when the None button is clicked. |
| System.DateTime GetCalendarValue(); | Returns the current value in the DateTimePickerAdv control. |

**DockingManager**

| Method | Description |
|---|---|
| DockStateChange(string dock,string prevState, string ctrl,string hostForm,string dockingStyle) | Changes the docking window according to the specified current and previous state (i.e Pinned, Unpinned, Tabbed, and MDIChild). |
| VisibilityChange(string ctrlName,string visibility) | Changes the visibility of the docked control according to the specified state. |
| ActivateControl(string ctrlName) | Activates the specified control. |
| FloatStateChange(string ctrlName, string x, string y, string width, string height) | Changes the state of the docking window into a floating state with the specified location and size. |

**GroupBar**

| Method | Description |
|---|---|
| SelectGroup(object index, string itemText) | Selects the GroupBar item. |
| DropDownButtonClick() | Simulates click in the Navigation pane drop-down button. |

### GroupView

| Method | Description |
|---|---|
| SelectItem(object item) | Selects the GroupView item. |
| DropItem(int index, object source) | Drag and drop the GroupView item. |

### MultiColumnComboBox

| Method | Description |
|---|---|
| DropDown() | Shows the hidden grid in the MultiColumnComboBox. |
| SelectIndex(int index) | Selects the given index. |

### PopupMenu

| Method | Description |
|---|---|
| Select(string barText) | Selects the item from the pop-up menu. |

### ProgressBarAdv

| Method | Description |
|---|---|
| SetValue(int value) | Assigns the Progress bar value. |
| int GetValue() | Gets the current value of the ProgressBar. |

### RadioButtonAdv

| Method | Description |
|---|---|
| Set() | Changes Checked property to true. |
| bool IsSet() | Shows whether the RadioButtonAdv is set. |

### RibbonControlAdv

| Method | Description |
|---|---|
| RibbonMenuButtonClick() | Clicks the Ribbon menu button. |
| SelectRibbonMenuItem(object item) | Selects the ribbon menu item. |
| Close() | Closes the parent form of RibbonControlAdv. |
| Activate() | Activates the parent form of RibbonControlAdv. |
| Maximize() | Maximizes the parent form of RibbonControlAdv. |
| Minimize() | Minimizes the parent form of RibbonControlAdv. |
| Restore() | Restores the parent form of RibbonControlAdv. |
| SelecTab(object tabItem) | Selects the Ribbon Tab item. |
| MinimizingPanel() | Minimizes the Ribbon Tab panel. |
| MaximizingPanel() | Maximizes the Ribbon Tab panel. |

### ScrollerFrame

| Method | Description |
|---|---|
| ScrollValue(int value) | The position of the scroll to be specified. |

### TabbedMDIManager

| Method | Description |
|---|---|
| ClosePage(object tabPage) | Closes the specified tab page. |
| SelectPage(object tab) | Selects the specified tab page. |

### TabControlAdv

| Method | Description |
|---|---|
| SelectPage(object tab) | Selects the tab page in the TabPageAdv control. |
| RightClick(object tab) | Performs a rightclick on the tab page in the TabPageAdv control. |
| ClosePage(object tab) | Closes the tab page in the TabPageAdv control. |

**XPTaskBar**

| Method | Description |
|---|---|
| Expand(string headerText) | Expands the content area of the task bar box. |
| Collapse(string headerText) | Collapses the content area of the task bar box. |
| ItemClick(string headerText, string itemText) | Performs click on the item described in the tag. |
| **Helper Functions** | |
| string GetTag(int itemIndex) | Retrieves the tag information for the given item index. |
| string GetHeaderText() | Retrieves the group or header text of the task bar box being called. |
| string GetItemText(int itemIndex) | Retrieves the item text for the given item index from the task bar box called |
| int GetTaskBarBoxCount() | Gets the number of task bar boxes in the XPTaskBar. |
| int GetExpandedTaskBarBoxCount() | Gets the number of expanded task bar boxes. |
| int GetCollapsedTaskBarBoxCount() | Gets the number of collapsed task bar boxes. |
| bool FindItem(string itemText, out string headerText, out int itemIndex); | Helps to find an item’s existence. |

**TextBoxExt**

| Method | Description |
|---|---|
| Set(string text) | Sets the text in the TextBoxExt. |
| SelectText(string selText, object start, object length); | Select the text in the TextBoxExt. |

**ThemedCheckButton**

| Method | Description |
|---|---|
| Set(string chkState) | Sets the CheckState of the CheckBox in the DateTimeAdv. |
| string GetCheckState() | Gets the CheckState of the CheckBox in the DateTimeAdv |

**TreeViewAdv**

| Method | Description |
|---|---|
| CollapseNode(string fullPath) | Collapses the specified node. |
| ExpandNode(string fullPath) | Expands the specified node. |
| SetCheckState(string fullPath, string checkState) | Sets the specified state of the CheckBox/OptionButton for the specified node. |
| SelectNodeWithModifierKeys(string fullPath,string ctrl, string shift) | Selects the specified node according to the selection mode. |
| BackupNodeDetails(string fullPath) | Backs up the node details before editing. |
| EditNode(string nodeText) | Edits the specified node. |
| DragDrop(string fullPath) | Perform the drag and drop operation for the nodes in the SelectedNodes list, which is added during drag over event. |
| AddToSelectedNodeList(string fullPath) | Adds the specifed node into selected node list during Drag over event. |
| DoubleClick(string fullPath) | Handles the double-click event of TreeViewAdv. |

| | |
|---|---|
| SelectNode(string fullPath) | Selects the node in SingleSelect mode. |
| RMouseDown(int x, int y) | Performs a right mouse click. |
| TreeNodeAdv GetNodeFromPath(string fullPath) | Gets the tree node from the path. |
| Point GetPointFromNode(TreeNodeAdv node) | Returns the TextBounds point of the specified node. |
| RightClickNode(string fullPath) | Right-clicks the specified node. |

**XPMenus**

| Method | Description |
|---|---|
| Select(string text) | Performs click on a bar item. |
| string TraceParentRoot(string barItemText) | For the given text of the required menu, TraceParentRoot will retrieve the full path as recoded. |
| int MenuItemPos(string ParentText, string barItemText) | For the given text of the required menu, MenuItemPos will return the position of the menu item. |

**XPToolBar**

| Method | Description |
|---|---|
| Select(string ID) | Performs click in the barItem. |
| Popup(string ID) | Shows the popup of the parent bar item. |

**XPTaskBar**

| Method | Description |
|---|---|
| Expand(string headerText) | Expands the content area of the task bar box. |
| Collapse(string headerText) | Collapses the content area of the task bar box. |
| ItemClick(string headerText, string itemTag) | Performs a click in the item described in the tag. |

| Helper Functions | |
|---|---|
| string GetTag(int itemIndex) | Retrieves the tag information for the given itemIndex. |
| string GetHeaderText() | Retrieves the group/header text of the task bar box called from. |
| string GetItemText(int itemIndex) | Retrieves the item text for the given item index from the task bar box called. |
| int GetTaskBarBoxCount() | Number of task bar boxes. |
| int GetExpandedTaskBarBoxCount() | Number of expanded task bar boxes. |
| GetCollapsedTaskBarBoxCount() | Number of collapsed task bar boxes. |
| bool FindItem(string itemText, out string headerText, out int itemIndex) | Helps to find if an item exists. |

**SplitContainerAdv**

| Method | Description |
|---|---|
| MoveSplitter(int distance) | Adjusts the distance of the splitter. |
| CollapsePanel(string collapse) | Collapses the panel. |

**TabSplitterContainer**

| Method | Description |
|---|---|
| Collapse(string collapse) | Collapses the pane to the bottom. |
| ChangeOrientation(string orientation) | Changes the orientation. |
| MoveSplitter(int position) | Adjusts the position of the splitter. |
| SwapPanes(string swap) | Swaps primary and secondary panes. |
| SelectPrimaryTab(int index) | Selects the primary tab page based on the given index. |

| | |
|---|---|
| SelectSecondaryTab(int index) | Selects the secondary tab page based on the given index. |

**TrackBarEx**

| Method | Description |
|---|---|
| SetValue(int value) | Sets the value. |

**RangeSlider**

| Method | Description |
|---|---|
| SetValue(int min, int max) | Sets the values. |

**NavigationView**

| Method | Description |
|---|---|
| Select(int x, int y) | Clicks the specified x and y value. |
| ActivateBar(string barName) | Selects the bar based on the given name. |

**TabBarSplitterControl**

| Method | Description |
|---|---|
| Select(string tab) | The name of the selected tab. |
| SetSplitterPosition(string tab, int vSplit, int hSplit) | The splitter position in the tab bar page. |
| string GetTabName(int index) | The label in the tab page can be found by passing the index. |

## 4.3 Essential Chart

The following are the recorded methods and their corresponding descriptions for Essential Chart:

**ChartControl**

| Method | Description |
|---|---|
| RegionClick(double x, double y) | The point on the chart region to be clicked. |
| RegionRightClick(double x, double y) | The point on the chart region to be right-clicked. |
| RegionDoubleClick(double x, double y) | The point on the chart region to be double-clicked. |
| TitleClick(int x, int y) | The region on the title to be clicked. |
| LegendClick(int x, int y) | The region on the legend to be clicked. |
| SetItemCheckState(string itemtext, string checkstate) | Setting the legend item check box. |
| SetLegendFloatingLocation(int x, int y) | The location of the legend if it is floating. |
| SetLegendNonFloatingLocation(object pos, object align); | The location of the fixed position in or on the QTP. |
| SetTitleFloatingLocation(int x, int y) | The location of the legend if it is floating. |
| SetTitleNonFloatingLocation(object pos, object align) | The location of the fixed position in or on the QTP. |
| ZoomXAxis(object min, object max) | The values of X-coordinates to zoom the chart. |
| ZoomYAxis(object min, object max) | The values of Y-coordinates to zoom the chart. |
| int GetSeriesCount(); | Gets the count of series within the chart. |
| int GetPointsCount(int series); | The point count on the specified series. |
| double GetMaxYValue(int series, int point); | The maximum Y-value of the specified point. |
| double GetXvalue(int series, int point); | The X-value of the specified point. |
| string GetChartType(int series); | Gets the type of the chart. |
| string GetXAxisText(); | Gets the text that appeared on the X-axis. |
| string GetYAxisText(); | Gets the text that appeared on the Y-axis. |

## 4.4 Essential Schedule

The following are the recorded methods and their corresponding descriptions for Essential Schedule:

**Schedule Control**

| Method | Description |
|---|---|
| DblClick(int row, int col) | Double-click a schedule row. |
| RightClick(int row, int col) | Right-click a schedule row. |
| TimeDrag (int row, int col, object newStartTime, object newEndTime) | Adjust the timeline for an appointment. |
| ItemDrag (int row, int col, object newStartTime, object newEndTime) | Move appointment to some other timeline. |
| Scroll(int value) | Scroll the schedule control. |

## 4.5 Essential Diagram

The following are the recorded methods and their corresponding descriptions for Essential Diagram:

**Diagram Control**

| Method | Description |
|---|---|
| ConnectNodes(string startNode, string endNode, string connector) | Connects the specified nodes using the connector. |
| SelectNode(string name) | Selects a diagram node. |
| DblClick(string name) | Double-clicks  a diagram node. |
| RotateNode(string node, float offset) | Rotates a diagram node to the given offset. |
| ResizeNode(string node, float offsetX, float OffsetY) | Resizes a diagram node to the given offset. |
| MoveNode(string node, float offsetX, float OffsetY) | Moves a diagram node to a new location. |
| Zoom(float magnification) | Zoom the diagram view. |

| | |
|---|---|
| Scroll(double x, double y) | Scroll the diagram view. |

# 5 Known Issues

The following are the known issues in various platforms that are yet to be solved.

## 5.1 General

Documentation column is not supported in the Keyword View.

## 5.2 Essential Grid

Grid does not support drop-down controls such as Combo box, Grid List control, and so on.

## 5.3 Essential Tools

The following are the list of tools with their respective known issues:

**Group Bar**

When the Stacked Mode is set to true, the NavigationPanelButtonClick is not recorded.

**GroupView**

When the button view is set to false, the drag-and-drop, or re-ordering, of the GroupView item is not recorded. On clicking the re-ordered item, the index is recorded correctly.

**DateTimePickerAdv**

1. The events on the header panel that are inside the pop-up window cannot be replayed. The SetCurrentCell and ResizeRow events of the Syncfusion.QuickTestProfessional.Grid that are associated with the Calendar are triggered by the pop-up window. These events are recorded, but cannot be played back in the replay. While replaying, they should be manually removed.
2. Once the calendar events are handled, the replay works slower. This is because of the 'for each' loop in the replay, which enables you to trace all the controls that are inside the pop-up window and then show or hide them as you need.

### Docking Manager

1. When two controls are in tabbed style and you click on the inactive tab and drag it outside the tabbed mode, it will not replay properly. In order to avoid this problem, select the tab that is going to be dragged, click on the tab, and drag it outside.
2. When a floating form state is changed to an MDIChild state and MDIChild state is changed to a floating form state, it will not replay properly.

### Ribbon Control

The Quick access panel customize menu will not be recorded.

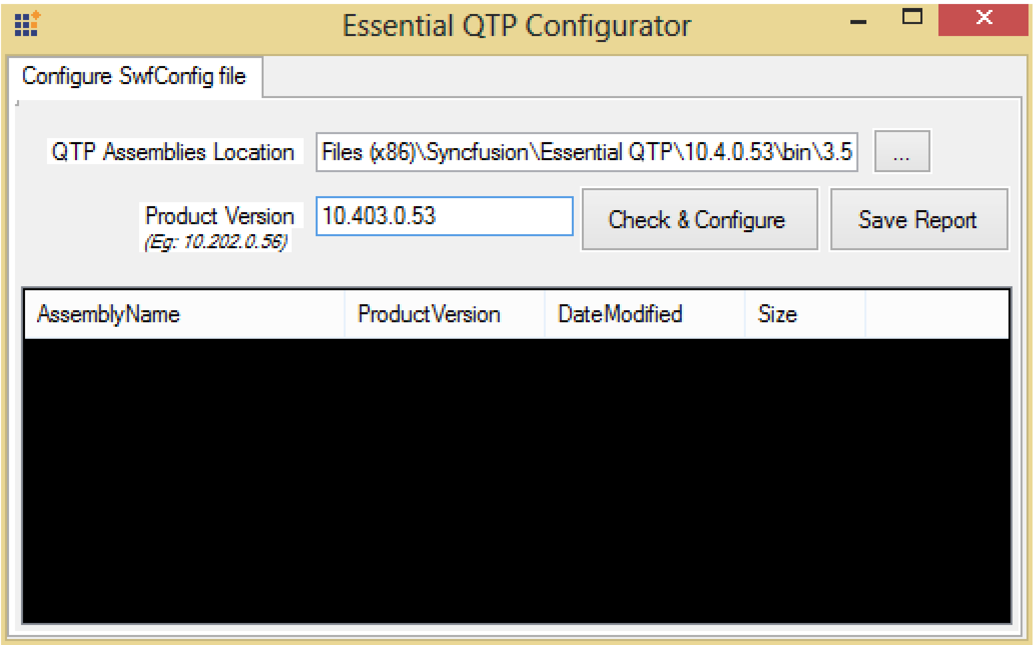
# 6 Utilities

## 6.1 Configuring the SwfConfig file

An XML file in QTP called **SwfConfig** is the configuration file located at **(Installed location of Essential QuickTest Professional)\Config\<version-2.0, 3.5, or 4.0>\swfconfig**, which contains all the mapping information for QTP to recognize Syncfusion controls. Using the SwfConfig utility, users can easily configure the **SwfConfig.xml** file in HP QTP.

**Steps to Configure the SwfConfig.xml File**

1. Open the Syncfusion Essential QTP Configurator located at **(Installed location of Essential QuickTest Professional)\Utilities\SwfConfigUtility\SwfConfigUtility.exe**. Enter the QTP assemblies' location in the **QTP Assemblies Location** textbox and the Essential Studio version with framework in the **Product Version** textbox. After entering the details, click **Check & Configure**. It will create the **swfconfig.xml** file for that particular version. Refer to the following image.

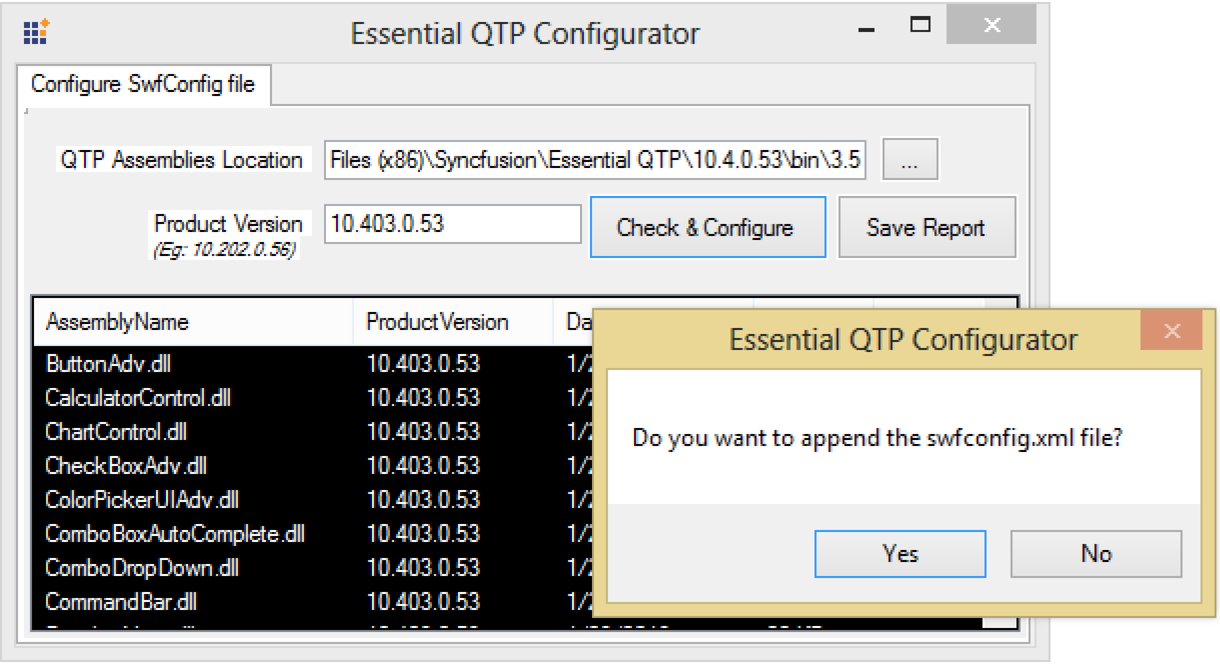


QTP Assemblies Location: Files (x86)\Syncfusion\Essential QTP\10.4.0.53\bin\3.5

Product Version (Eg: 10.202.0.56): 10.403.0.53

*Figure 29: Creating the SwfConfig.xml File for Essential Studio 10.3*

2. Then Essential QTP Configurator shows the dialog box for appending the swfconfig.xml file. Click **Yes** to append the swfconfig.xml file in the QTP machine.



Essential QTP Configurator

Configure SwfConfig file

QTP Assemblies Location: Files (x86)\Syncfusion\Essential QTP\10.4.0.53\bin\3.5

Product Version (Eg: 10.202.0.56): 10.403.0.53

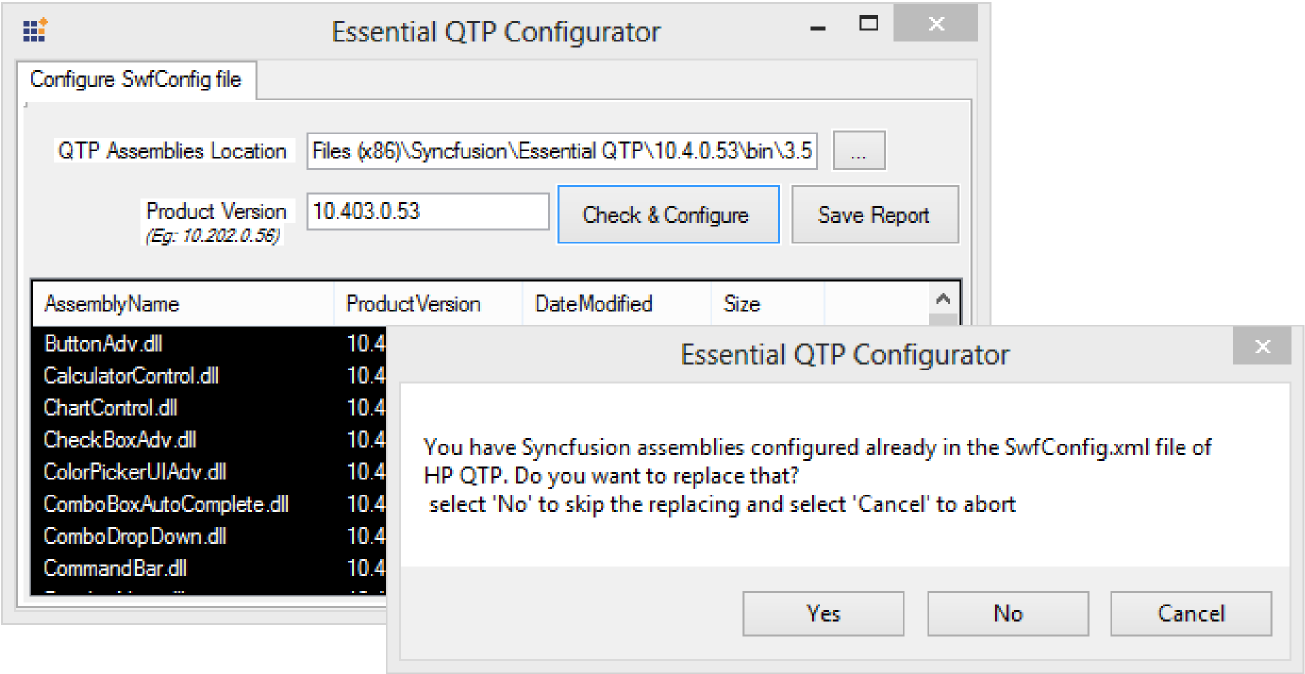
| AssemblyName | ProductVersion | Da |
|---|---|---|
| ButtonAdv.dll | 10.403.0.53 | 1/ |
| CalculatorControl.dll | 10.403.0.53 | 1/ |
| ChartControl.dll | 10.403.0.53 | 1/ |
| CheckBoxAdv.dll | 10.403.0.53 | 1/ |
| ColorPickerUIAdv.dll | 10.403.0.53 | 1/ |
| ComboBoxAutoComplete.dll | 10.403.0.53 | 1/ |
| ComboDropDown.dll | 10.403.0.53 | 1/ |
| CommandBar.dll | 10.403.0.53 | 1/ |

Essential QTP Configurator

Do you want to append the swfconfig.xml file?

*Figure 30: Appending the SwfConfig File*

3. If your system already has a swfconfig.xml file, then another dialog box will appear asking to replace the existing swfconfig.xml. Click **Yes** to replace the old swfconfig.xml file with the current framework swfconfig.xml file on your machine. If you want to keep both files in the same folder, click **No**.



QTP Assemblies Location: Files (x86)\Syncfusion\Essential QTP\10.4.0.53\bin\3.5

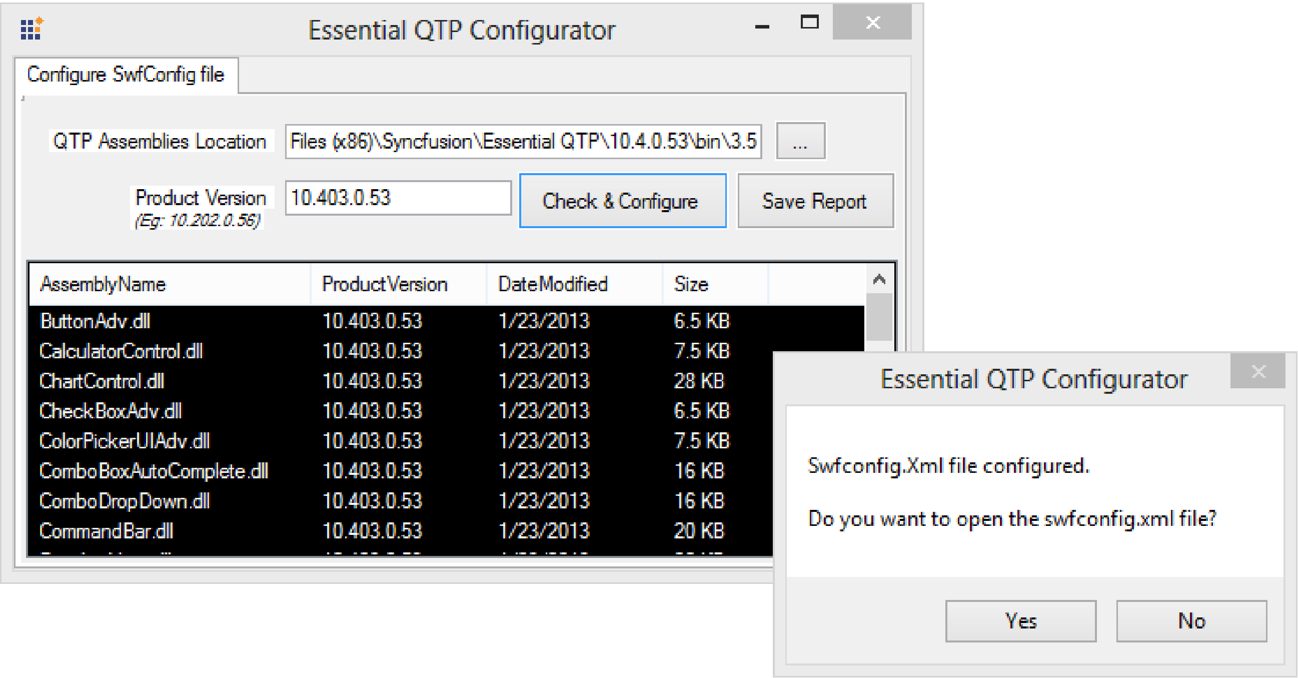
Product Version (Eg: 10.202.0.56): 10.403.0.53

| AssemblyName | ProductVersion | DateModified | Size |
|---|---|---|---|
| ButtonAdv.dll | 10.4 | | |
| CalculatorControl.dll | 10.4 | | |
| ChartControl.dll | 10.4 | | |
| CheckBoxAdv.dll | 10.4 | | |
| ColorPickerUIAdv.dll | 10.4 | | |
| ComboBoxAutoComplete.dll | 10.4 | | |
| ComboDropDown.dll | 10.4 | | |
| CommandBar.dll | 10.4 | | |

You have Syncfusion assemblies configured already in the SwfConfig.xml file of HP QTP. Do you want to replace that?
select 'No' to skip the replacing and select 'Cancel' to abort

*Figure 31: Replacing the SwfConfig.xml File*

4. After generating the swfconfig.xml file, the system will ask whether you want to open it. Click **Yes** to save and open the new swfconfig.xml file.



QTP Assemblies Location: Files (x86)\Syncfusion\Essential QTP\10.4.0.53\bin\3.5

Product Version (Eg: 10.202.0.56): 10.403.0.53

| AssemblyName | ProductVersion | DateModified | Size |
|---|---|---|---|
| ButtonAdv.dll | 10.403.0.53 | 1/23/2013 | 6.5 KB |
| CalculatorControl.dll | 10.403.0.53 | 1/23/2013 | 7.5 KB |
| ChartControl.dll | 10.403.0.53 | 1/23/2013 | 28 KB |
| CheckBoxAdv.dll | 10.403.0.53 | 1/23/2013 | 6.5 KB |
| ColorPickerUIAdv.dll | 10.403.0.53 | 1/23/2013 | 7.5 KB |
| ComboBoxAutoComplete.dll | 10.403.0.53 | 1/23/2013 | 16 KB |
| ComboDropDown.dll | 10.403.0.53 | 1/23/2013 | 16 KB |
| CommandBar.dll | 10.403.0.53 | 1/23/2013 | 20 KB |

Swfconfig.Xml file configured.

Do you want to open the swfconfig.xml file?

*Figure 32: Opening the new SwfConfig.xml File*

5. Restart QTP once the SwfConfig.xml file is saved to refresh the mappings to the required controls before starting the test.

# 7 Frequently Asked Questions

## 7.1 General

### 7.1.1 How to manually configure Syncfusion control to work with QTP

**Steps to Configure QTP to use the Custom Libraries shipped in Essential QuickTest Professional**

1. Navigate to the following path:

   **(Installed location of Essential QuickTest Professional)\Config**

***Note: You will find three folders here: 2.0, 3.5 and 4.0. The folders 2.0, 3.5 and 4.0 consist of swfconfig files for .NET 2.0, .NET 3.5 and .NET 4.0 frameworks respectively.***

2. Open the swfconfig file by double-clicking it. You can view the mapping for all the supported controls here. Given below is the sample code that maps the grid control to its corresponding DLL.

```
[XML]
       <CC <Control Type="Syncfusion.Windows.Forms.Grid.GridControl">
 <CustomRecord>
 <Component>
 <Context>AUT</Context>
 <DllName>C:\Program files\Syncfusion\Essential TestStudio\<Version
Number>\Bin\2.0\GridControl.dll</DllName>
 <TypeName>Syncfusion.TestStudio.Grid.GridControl</TypeName>
 </Component>
  </CustomRecord>
 <CustomReplay>
 <Component>
 <Context>AUT</Context>
 <DllName>C:\Program files\Syncfusion\Essential TestStudio\<Version
number>\Bin\2.0\GridControl.dll</DllName>
 <TypeName>Syncfusion.TestStudio.Grid.GridControl</TypeName>
   </Component>
 </CustomReplay>
</Control>
```

***Note: In the preceding code, the fully qualified name of the DLL given in the <DllName> tag assumes that you have installed the Essential QuickTest Professional in the following default path:***

***C:\Program Files\Syncfusion\Essential QuickTest Professional\<Version number>***

If you have installed Essential QuickTest Professional in any other path, you need to give the correct path of the DLL in all the <DllName> tag. For example, if Essential QuickTest Professional is located in D:\Essential QuickTest Professional\<version number>, then the code should be as follows:

```
[XML]

 <Control Type="Syncfusion.Windows.Forms.Grid.GridControl">
<CustomRecord>
<Component>
<Context>AUT</Context>
<DllName>D:\Essential TestStudio\<Version
Number>\Bin\2.0\GridControl.dll</DllName>
<TypeName>Syncfusion.TestStudio.Grid.GridControl</TypeName>
    </Component>
 </CustomRecord>
 <CustomReplay>
 <Component>
 <Context>AUT</Context>
 <DllName>D:\Essential TestStudio\<Version
number>\Bin\2.0\GridControl.dll</DllName>
 <TypeName>Syncfusion.TestStudio.Grid.GridControl</TypeName>
   </Component>
 </CustomReplay>
  </Control>
```

3. Select the segment of the code containing the controls to be tested.
4. On the **Edit** menu, click **Copy**.

***Note: While selecting the code for copying, exclude the following lines of code:***

```
[XML]

<?xml version="1.0" encoding="UTF-8" ?>
```

5. Open the SwfConfig.xml file located under the following location:
***<QuickTest Professional Installation Path>\dat\SwfConfig.xml***

6. Paste the copied segment under the <?xml> tag in SwfConfig.xml.

***Note: The SwfConfig.xml file will look like the following:***

```
[XML]

<?xml version="1.0" encoding="UTF-8" ?>
 <Controls>
  <Control Type="Syncfusion.Windows.Forms.Grid.GridControl">
 <CustomRecord>
 <Component>
 <Context>AUT</Context>
 <DllName>C:\Program files\Syncfusion\Essential TestStudio\<Version
Number>\Bin\2.0\GridControl.dll</DllName>
 <TypeName>Syncfusion.TestStudio.Grid.GridControl</TypeName>
      </Component>
 </CustomRecord>
<CustomReplay>
<Component>
<Context>AUT</Context>
<DllName>C:\Program files\Syncfusion\Essential TestStudio\<Version
number>\Bin\2.0\GridControl.dll</DllName>
<TypeName>Syncfusion.TestStudio.Grid.GridControl</TypeName>
      </Component>
 </CusmReplay>
</Control>
  ..........
</Controls>
```

***Note: Ensure that the element <DllName> contains the correct path to the corresponding DLL.***

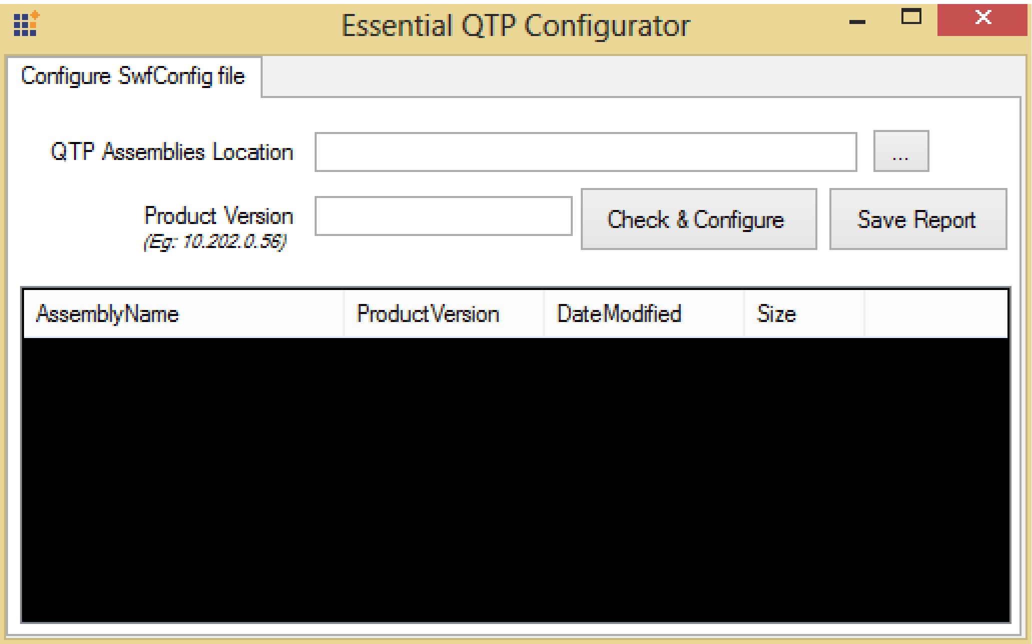
7. Save the SwfConfig.xml file.

8. Restart QTP once the SwfConfig.xml file is saved to refresh the mappings to the required controls before starting the test.

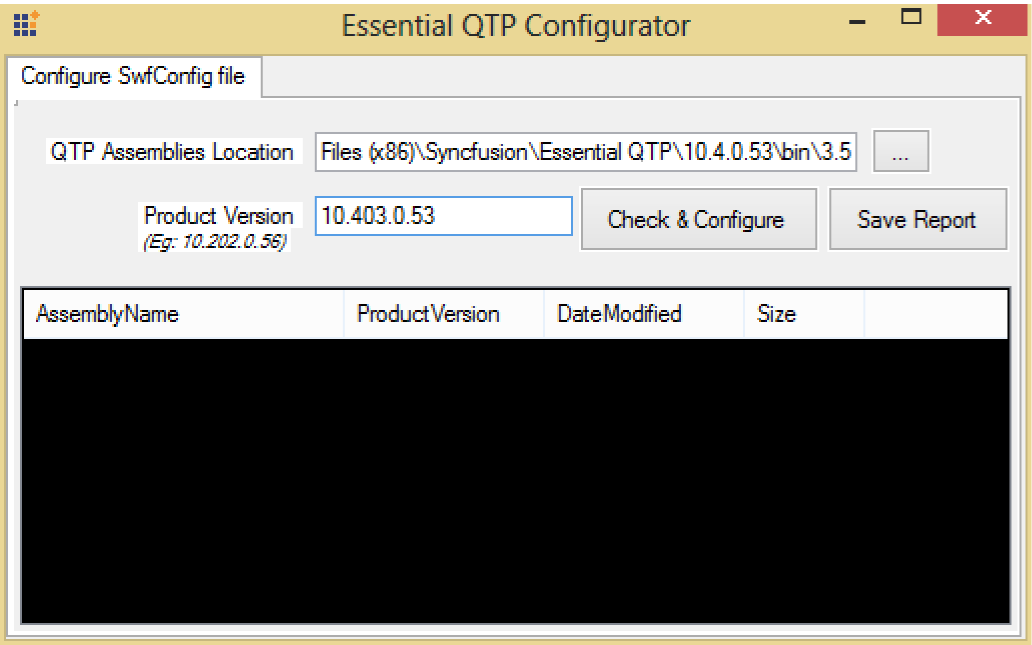
***Note: Mapping for the required controls can be done in a similar manner.***

### 7.1.2 How to know whether my swfconfig file holds an invalid assembly path reference

1. Open the Syncfusion Essential QTP Configurator located at **(Installed location of Essential QuickTest Professional)\Utilities\SwfConfigUtility\SwfConfigUtility.exe.**



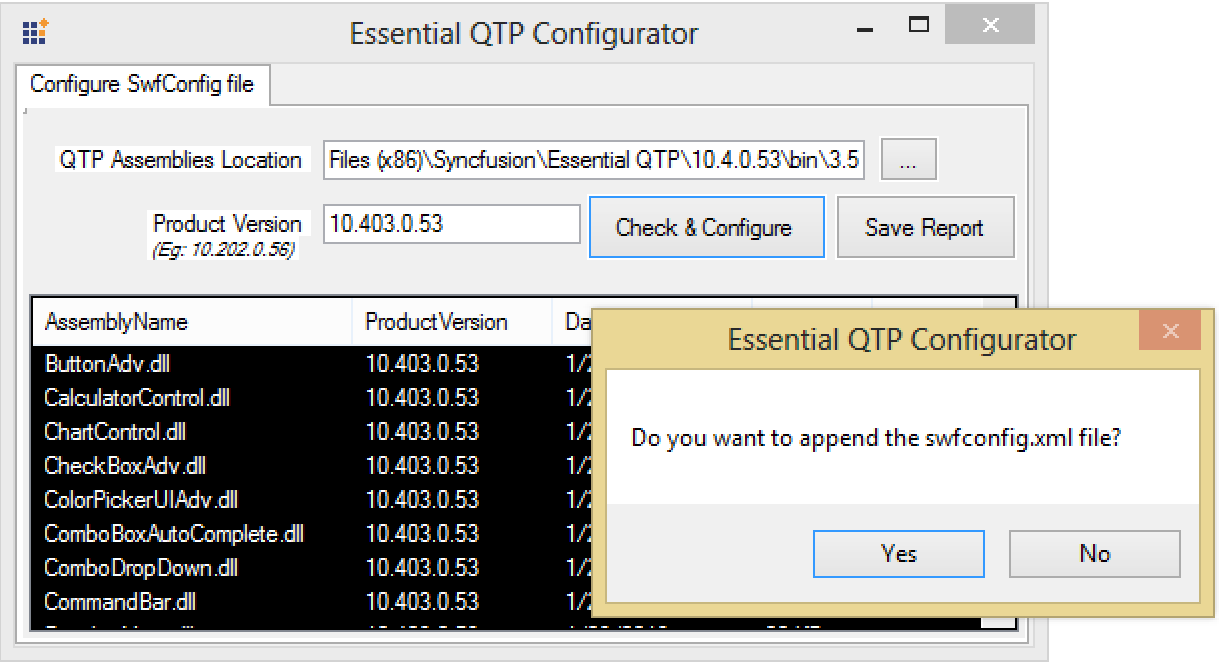
2. Enter the QTP assemblies' location in the **QTP Assemblies Location** textbox and the Essential Studio version with framework in the **Product Version** textbox.



QTP Assemblies Location: Files (x86)\Syncfusion\Essential QTP\10.4.0.53\bin\3.5

Product Version (Eg: 10.202.0.56): 10.403.0.53

3. After entering the details, click **Check & Configure.**
4. If the swfconfig file holds the valid reference path, then the swfconfig utility shows the dialog box to save the swfconfig.xml file.



QTP Assemblies Location: Files (x86)\Syncfusion\Essential QTP\10.4.0.53\bin\3.5

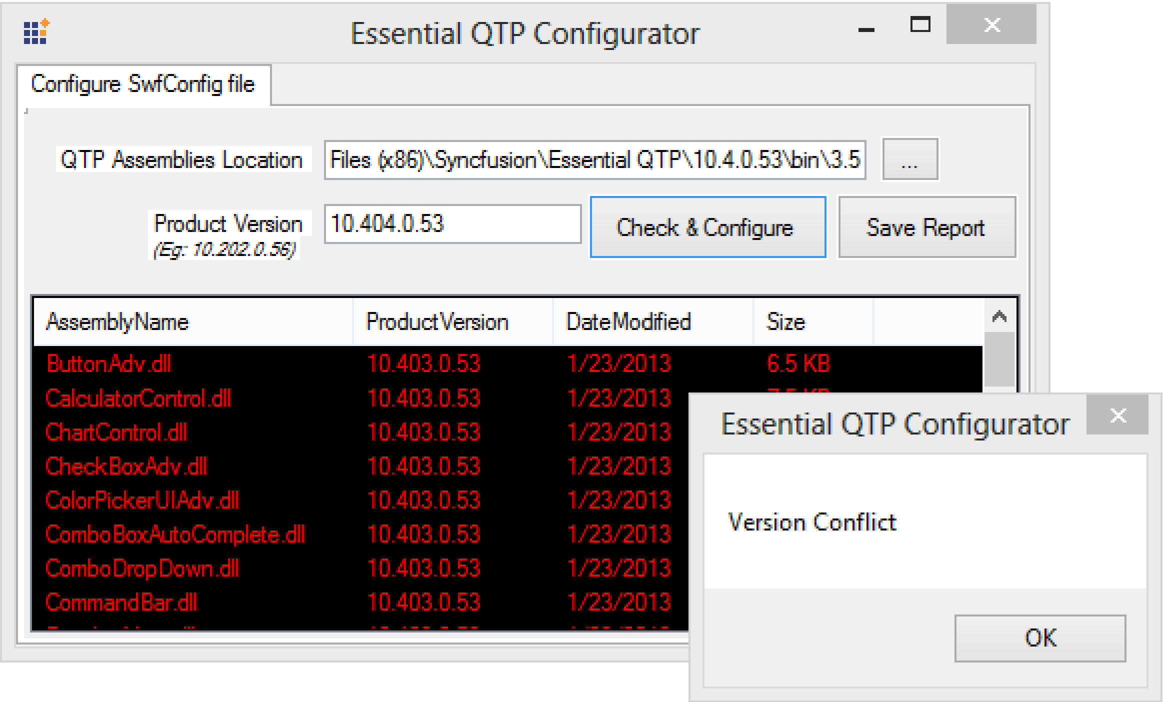
Product Version (Eg: 10.202.0.56): 10.403.0.53

| AssemblyName | ProductVersion | Da |
|---|---|---|
| ButtonAdv.dll | 10.403.0.53 | 1/ |
| CalculatorControl.dll | 10.403.0.53 | 1/ |
| ChartControl.dll | 10.403.0.53 | 1/ |
| CheckBoxAdv.dll | 10.403.0.53 | 1/ |
| ColorPickerUIAdv.dll | 10.403.0.53 | 1/ |
| ComboBoxAutoComplete.dll | 10.403.0.53 | 1/ |
| ComboDropDown.dll | 10.403.0.53 | 1/ |
| CommandBar.dll | 10.403.0.53 | 1/ |

Essential QTP Configurator

Do you want to append the swfconfig.xml file?

5. If you get the error message box with a version conflict error, the swfconfig file holds an invalid assembly reference path.



QTP Assemblies Location: Files (x86)\Syncfusion\Essential QTP\10.4.0.53\bin\3.5

Product Version (Eg: 10.202.0.56): 10.404.0.53

| AssemblyName | ProductVersion | DateModified | Size |
|---|---|---|---|
| ButtonAdv.dll | 10.403.0.53 | 1/23/2013 | 6.5 KB |
| CalculatorControl.dll | 10.403.0.53 | 1/23/2013 | |
| ChartControl.dll | 10.403.0.53 | 1/23/2013 | |
| CheckBoxAdv.dll | 10.403.0.53 | 1/23/2013 | |
| ColorPickerUIAdv.dll | 10.403.0.53 | 1/23/2013 | |
| ComboBoxAutoComplete.dll | 10.403.0.53 | 1/23/2013 | |
| ComboDropDown.dll | 10.403.0.53 | 1/23/2013 | |
| CommandBar.dll | 10.403.0.53 | 1/23/2013 | |

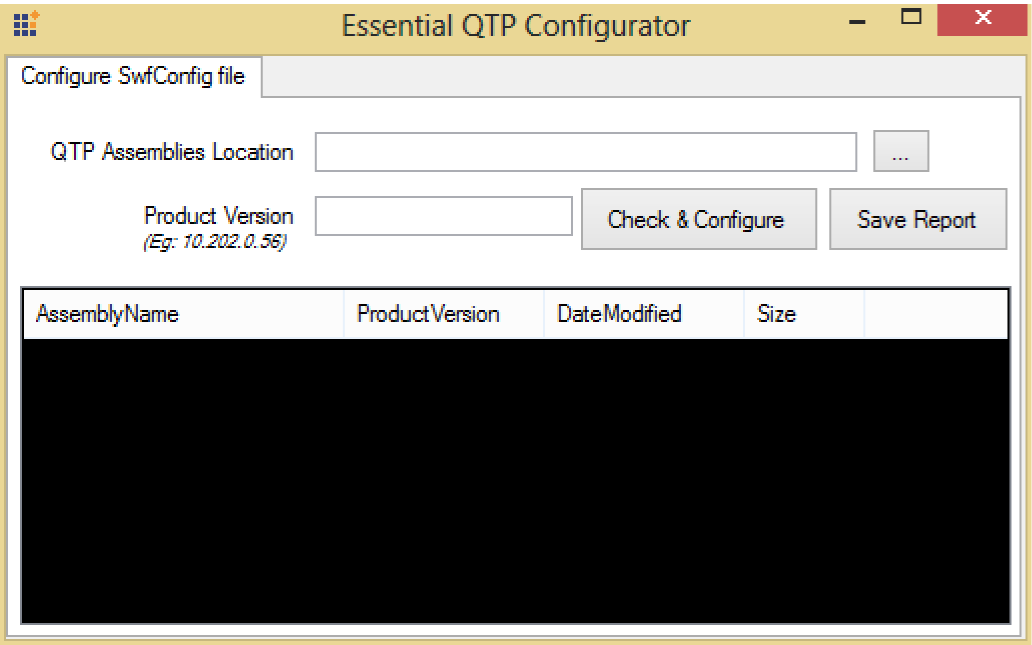
Essential QTP Configurator

Version Conflict

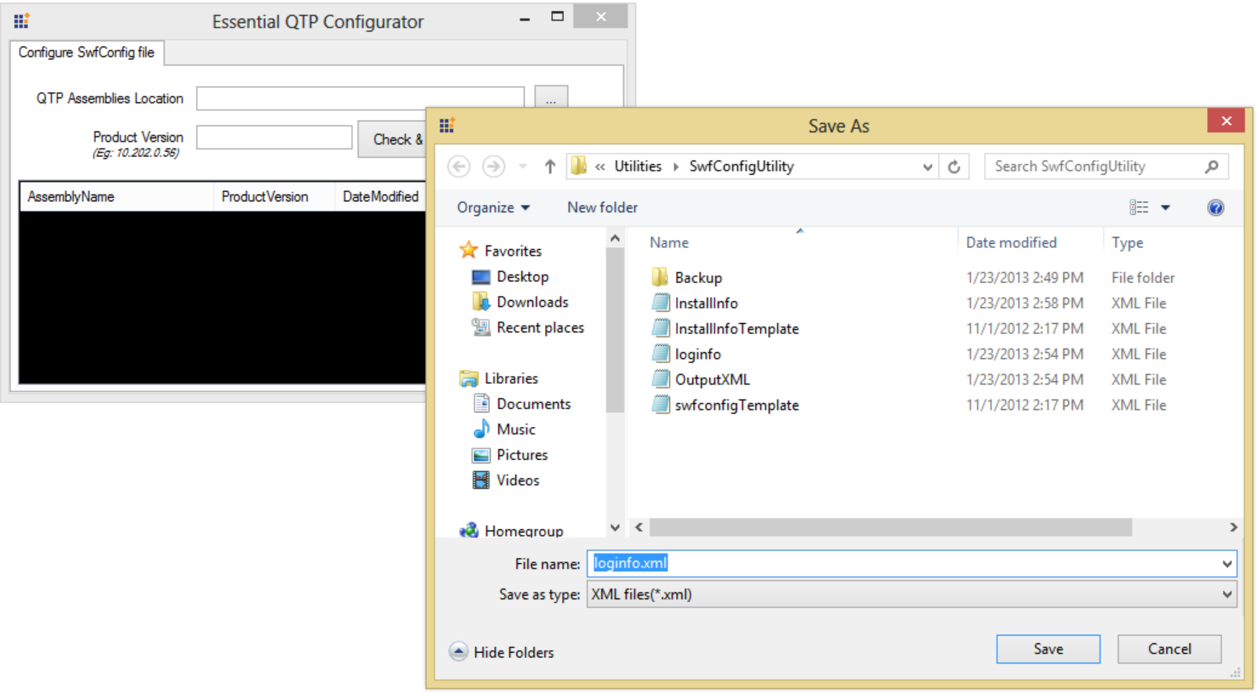
6. You can get the error details using the configuration file.

### 7.1.3 How to fetch installation information related to the Syncfusion QTP add-on

1. Open the Syncfusion Essential QTP Configurator located at **(Installed location of Essential QuickTest Professional)\Utilities\SwfConfigUtility\SwfConfigUtility.exe.**



2. Click the **Save Report** button to get the install information.
3. There is no need to enter the product version and install path details in the respective text boxes.
4. It will then show the save dialog box to save the install information.

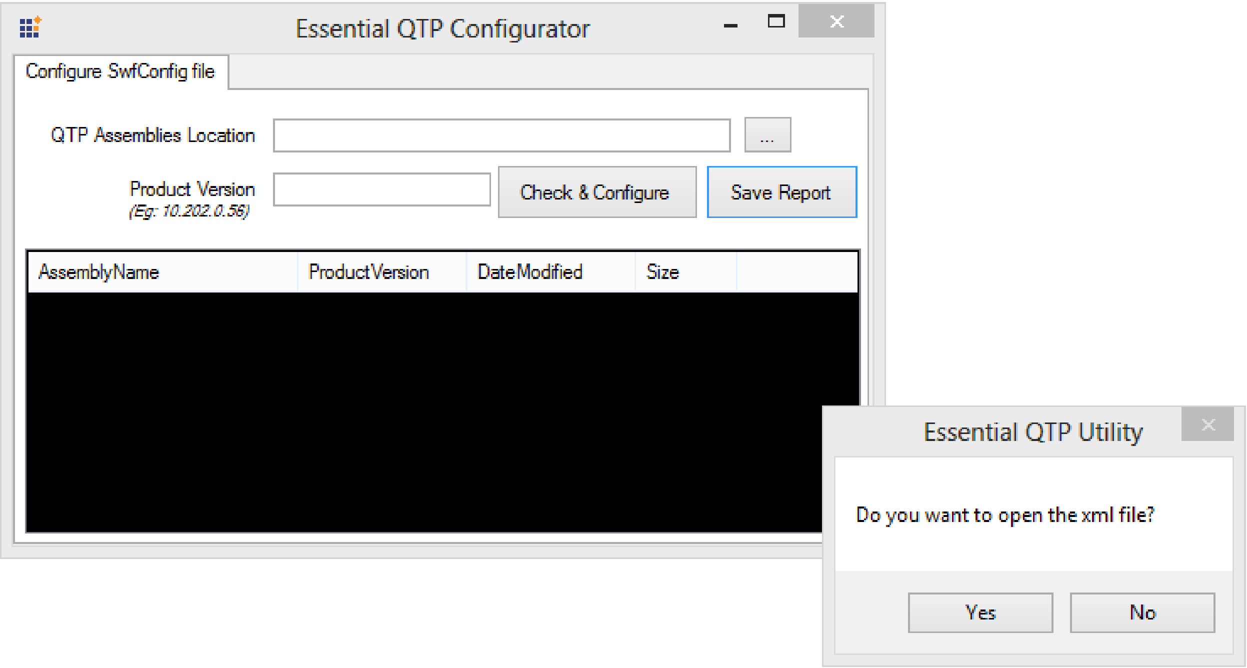


| Name | Date modified | Type |
|---|---|---|
| Backup | 1/23/2013 2:49 PM | File folder |
| InstallInfo | 1/23/2013 2:58 PM | XML File |
| InstallInfoTemplate | 11/1/2012 2:17 PM | XML File |
| loginfo | 1/23/2013 2:54 PM | XML File |
| OutputXML | 1/23/2013 2:54 PM | XML File |
| swfconfigTemplate | 11/1/2012 2:17 PM | XML File |

File name: loginfo.xml

Save as type: XML files(*.xml)

5. Then the dialog box is shown to open the xml file. Click yes if you want to read the xml file.



Do you want to open the xml file?

## 7.1.4 Why are Syncfusion controls not recognized even after adding the custom libraries?

The following are the troubleshooting steps to get the Syncfusion controls recognized in the QTP environment.

1. Make sure that the DLL path of the custom libraries is properly written in the SwfConfig.xml file. Refer to the Configuring Essential QuickTest Professional topic in this user guide for more details.

2. There are chances for typing errors to occur in the SwfConfig.xml. Ensure that there are no typing errors in the file and try replacing SwfConfig.xml at the correct location and restart QTP.

3. Sometimes the Syncfusion controls may not be recognized due to differences in the version numbers of Essential Studio and Essential QuickTest Professional, or .NET framework and Essential QuickTest Professional that are being used. Check if the version numbers of the assembly that is used to build the application and the Essential QuickTest Professional assembly are the same. If not, this can be solved by rebuilding the Custom Libraries with the required Syncfusion references and .NET framework. If you do not have the corresponding versions of Essential QuickTest Professional and Essential Studio, please contact us specifying the version of Essential QuickTest Professional that is required.

4. If the DLLs are the right version and are mapped correctly, and if the SwfConfig.xml is perfect, but there is still an issue of recognizing Syncfusion controls, then reinstall the .NET add-in for QTP. If the AUT (Application Under Test) is recorded as a WinObject (object in the Windows Environment), make a cross check with a small .NET application using a non-Syncfusion control to see if this control is also not recognized. If so, the problem is with the QTP or .NET add-in installed. Thus, we can isolate the problem with the .NET controls being recognized. SwfObject would be the right way to be recognized after the .NET add-in install.

## 7.1.5 How do I know that Essential QuickTest Professional works as expected?

When Syncfusion control events are recorded, they should be able to record with the method that is handled in the custom library (DLL). This will not occur if the mapping is not correct. If the mapping in the DllName tag of the SwfConfig.xml does not point to the required DLL, the recording would be seen as in the sample script below, which is a low-level recording already explained in the document.

```
[QTP]

SwfWindow("GridDataBoundGrid
CellTypes").SwfObject("gridDataBoundGrid2").SetCurrentCell 1,2
```

```
SwfWindow("GridDataBoundGrid 
CellTypes").SwfObject("gridDataBoundGrid2").SetCellData 1,2,"435.00"
SwfWindow("GridDataBoundGrid 
CellTypes").SwfObject("gridDataBoundGrid2").SetCurrentCell 2,1
SwfWindow("GridDataBoundGrid 
CellTypes").SwfObject("gridDataBoundGrid2").SetCellCheckBox 2,1,"ON"
SwfWindow("GridDataBoundGrid 
CellTypes").SwfObject("gridDataBoundGrid2").SetCurrentCell 3,1
SwfWindow("GridDataBoundGrid 
CellTypes").SwfObject("gridDataBoundGrid2").SetCellCheckBox 3,1,"OFF"
```

If the mapping points to the wrong version, then no scripts will be generated. The right version would be the same version as the AUT developed. For an example, if the AUT is developed in 7.3.0.20, the custom libraries (DLLs) should also be developed in 7.3.0.20. This means that Essential QuickTest Professional version 7.3.0.20 is required. With proper versions and proper mapping, the record will appear as shown in the script below:

```
[QTP]

SwfWindow("GridDataBoundGrid 
CellTypes").SwfObject("gridDataBoundGrid2").SetCurrentCell 1,2
SwfWindow("GridDataBoundGrid 
CellTypes").SwfObject("gridDataBoundGrid2").SetCellData 1,2,"435.00"
SwfWindow("GridDataBoundGrid 
CellTypes").SwfObject("gridDataBoundGrid2").SetCurrentCell 2,1
SwfWindow("GridDataBoundGrid 
CellTypes").SwfObject("gridDataBoundGrid2").SetCellCheckBox 2,1,"ON"
SwfWindow("GridDataBoundGrid 
CellTypes").SwfObject("gridDataBoundGrid2").SetCurrentCell 3,1
SwfWindow("GridDataBoundGrid 
CellTypes").SwfObject("gridDataBoundGrid2").SetCellCheckBox 3,1,"OFF"
```

In the above scripts, SetCurrentCell, SetCellData, and SetCellCheckBox are the methods of Grid control.

For the list of methods that will be recorded for all the supported controls, refer to the Supported Controls topic. You can also visit our Knowledge Base for Essential QuickTest Professional at the following link for more details: http://www.syncfusion.com/support/kb/tag/QTP

## 7.2 Essential Grid

## 7.2.1 How to get the description of the Check Box Cells and Normal Cells in Essential Grid

**Supported method to get the Description of GridCells**

The GetDescription method is used to get the description of the check box, push button, and normal cells in Essential Grid's record and reply process. This method returns the description of the check box, push button, and normal cells as well.

**Use Case Scenarios**

This feature enables you to get the description of checkbox, push button, and normal cells in QTP testing.

**Methods Table**

| Method | Description | Parameters | Return Type |
|---|---|---|---|
| GetDescription | Gets the description of grid cells for Essential Grid. | For the Grid control:<br>public object GetDescription(int row, int col) | Object |
| | | For the GridGrouping control:<br>public object GetDescription(object row, object col) | Object |

**Applying the GetDescription Method in QTP**

The following code examples illustrate how to use the **GetDescription** method.

```
[For GridControl]

SwfWindow("Form1").SwfObject("gridControl1").SetCurrentCell 3,1
MsgBox(SwfWindow("Form1").SwfObject("gridControl1").GetDescription(3,1))
```

```
[For GridGroupingControl]

SwfWindow("Form1").SwfObject("gridGroupingControl1").SetCurrentCell
3,"Col2"
MsgBox(SwfWindow("Form1").SwfObject("gridGroupingControl1").GetDescript
ion(5,"Col0"))
```

## 7.2.2 How to set the current cell in Grid

The SetCurrentCell method is used to set the current cell in Grid or activate a cell as the current cell in Grid. This method is used in GridControl, GridGroupingControl, and GridDataBoundControl.

**Use Case Scenarios**

This method enables a Grid cell in QTP testing.

**Methods Table**

| Method | Description | Parameters | Return Type |
|---|---|---|---|
| SetCurrentCell | Sets the location of current cell in Essential Grid | For the Grid control:<br>`public void SetCurrentCell(int row, int col)` | Void |
| | | For the GridGrouping control:<br>`public void SetCurrentCell(object row, object col)`<br>For GridDataBoundGrid control:<br>`public void SetCurrentCell(int row, int col)` | Void |

***Note: GridGrouping control uses iterated rows to set the current cell in the respective tables.***

The following code examples illustrate how to use the SetCurrentCell method.

```
[For GridControl]

SwfWindow("Form1").SwfObject("gridControl1").SetCurrentCell 3,1
```

```
[For GridGroupingControl]

SwfWindow("Form1").SwfObject("gridGroupingControl1").SetCurrentCell
3,"Col2"
```

## 7.3 Essential Tools

### 7.3.1 How to select the XPtool bar without ID

**Supported method to select the BarItem of XPToolBar**

The select method selects the BarItem of the XPToolBar. The click action is performed to select the BarItem of a tool.

**Use Case Scenarios**

This feature enables you to select the barItem of Xptools while clicking on a BarItem in QTP testing.

**Methods Table**

| Method | Description | Parameters | Return Type |
|---|---|---|---|
| Select | Select the BarItem of XPToolBar Essential Tools. | `public void Select(string ID)` | Void |

**Applying GetDescription Method in QTP**

The following code example illustrates how to use the **select** method.

```
SwfWindow("XPToolBarDemo").SwfObject("XPToolBar1").Select(“ID”))
```

### 7.3.2 How to check and uncheck the CheckBoxAdv

**Supported method to check status of CheckBoxAdv**

The GetCheckState method is used to find whether the checkBoxAdv is in checked or unchecked status. This method returns the answer in string.

**Use Case Scenarios**

This feature enables you to find whether the checkBoxAdv is checked or unchecked in QTP testing.

**Methods Table**

| Method | Description | Parameters | Return Type |
|---|---|---|---|

| | | | |
|---|---|---|---|
| GetCheckState | Get the check state of the CheckBoxAdv control in Essential Tools. | `public string GetCheckState()` | string |

**Applying GetCheckState Method in QTP**
The following code example illustrates how to use the GetCheckState method.

```
SwfWindow("QTPCheckBoxAdv").SwfObject("checkBox").GetCheckedState.Set “On”
MsgBox(SwfWindow("QTPCheckBoxAdv").SwfObject("checkBox").GetCheckedState())
```

## 7.3.3 How to collapse and expand the specified node in TreeViewADV

**Supported method to collapse and expand the specified node in TreeViewADV**
The CollapseNode method is used to collapse the specified node in TreeViewADV. The path of the node must be passed in the CollapseNode method. The ExpandNode method is used to expand the specified node in TreeViewADV.

**Use Case Scenarios**
This feature enables you to collapse and expand the specified node in TreeViewADV in QTP testing.

**Methods Table**

| Method | Description | Parameters | Return Type |
|---|---|---|---|
| CollapseNode | Collapse the specified node in TreeViewADV in Essential Tools. | `public void CollapseNode(string fullPath)` | Void |

| | | | |
|---|---|---|---|
| ExpandNode | ExpandNode the specified node in TreeViewADV in Essential Tools. | `public void ExpandNode (string fullPath)` | Void |

**Applying CollapseNode and ExpandNode Method in QTP**

The following code examples illustrate how to use the CollapseNode and ExpandNode method.

```
SwfWindow("QTPTreeViewAdv").SwfObject("Node1"). CollapseNode(“Node2”)
SwfWindow("QTPTreeViewAdv").SwfObject("Node1"). ExpandNode (“Node2”)
```

## 7.4 Essential Chart

### 7.4.1 How to get the displayed text in the X-axis and Y-axis

**Supported method to get the displayed text in the X-axis and Y-axis**

The GetXAxisText and GetYAxisText method is used to get to get the displayed text in the X-axis and Y-axis. This method returns the displayed text in string format.

**Use Case Scenarios**

This feature enables you to get the displayed text in the X-axis and Y-axis in QTP testing.

**Methods Table**

| Method | Description | Parameters | Return Type |
|---|---|---|---|
| GetXAxisText | Gets the displayed text of the X-axis in Essential Chart. | public string GetXAxisText() | String |
| GetYAxisText | Get the displayed text of the Y-axis in Essential Chart. | public string GetYAxisText() | String |

**Applying GetXAxisText and GetYAxisText Method in QTP**

The following code examples illustrate how to get the displayed text.

```
MsgBox(SwfWindow("ChartDemo").SwfObject("chart1"). GetXAxisText ())
MsgBox(SwfWindow("ChartDemo").SwfObject("chart1"). GetYAxisText ())
```

## 7.4.2 How to find the count of a series within the chart

**Supported method to find the series count within the chart.**

The GetSeriesCount method is used to get the series count within the chart. This method returns the displayed text in integer format.

**Use Case Scenarios**

This feature enables you to get the count of a series within the chart in QTP testing.

**Methods Table**

| Method | Description | Parameters | Return Type |
|---|---|---|---|
| GetSeriesCount | Gets the series count within the chart in Essential Chart. | public in GetSeriesCount() | Int |

**Applying GetSeriesCount in QTP**

The following code example illustrates how to get the series count in the chart.

```
[For Chart Control]
MsgBox(SwfWindow("ChartDemo").SwfObject("chart1"). GetSeriesCount ())
```

## 7.4.3 How to find the maximum Y-axis value in the specified region

**Supported method to find the maximum Y-axis value in the specified region**

The GetMaxYValue method is used to get the displayed maximum value in the Y-axis. This method returns the displayed value in double format.

### Use Case Scenarios

This feature enables you to get the maximum Y-axis value of a specified region in QTP testing.

### Methods Table

| Method | Description | Parameters | Return Type |
|---|---|---|---|
| GetMaxYValue | Get the Maximun Y axis value of specified region in Essential Chart. | public double GetMaxYValue(int series, int point) | double |

### Applying GetMaxYValue Method in QTP

The following code example illustrates how to get the displayed text.

```
[For Chart Control]

MsgBox(SwfWindow("ChartDemo").SwfObject("chart1"). GetMaxYValue (10,2))
```

## 7.5 Essential Schedule

### 7.5.1 How to reschedule the appointment to another timeline

**Supported method to reschedule the appointment to another timeline in the schedule control**

The ItemDrag method is used to reschedule the appointment to another timeline in the schedule control. The appointments are rescheduled to other dates based on the given start and end time.

### Use Case Scenarios

This feature enables you to reschedule the appointment to another timeline in the schedule control in QTP testing.

### Methods Table

| Method | Description | Parameters | Return Type |
|---|---|---|---|

| | | | |
|---|---|---|---|
| ItemDrag | Reschedule the appointment to another timeline in schedule control. | public void ItemDrag(string apptSubject, string oldStartTime, string oldEndTime, string newStartTime, string newEndTime) | void |

**Applying ItemDrag in QTP**

The following code example illustrates how to reschedule the appointment in schedule control.

```
[For Schedule Control]
SwfWindow("GridSchedulerDemo").SwfObject("Scheduler").
ItemDrag(“Appointment1”, “ 10/02/2012”, “14/02/2013”, “ 10/02/2013”,
“14/02/2014”)
```

## 7.5.2 How to reschedule the timeline of an appointment

**Supported method to reschedule the timeline of an appointment in the schedule control**

The TimeDrag method is used to reschedule the timeline of the appointment in the schedule control. The appointments are rescheduled to another time based on the new start time and new end time.

**Use Case Scenarios**

This feature enables you to reschedule the timeline of appointments in the schedule control in QTP testing.

**Methods Table**

| Method | Description | Parameters | Return Type |
|---|---|---|---|
| TimeDrag | Reschedule the timeline to another appointment in the schedule control. | public void TimeDrag(string apptSubject, string oldStartTime, string oldEndTime, string newStartTime, string newEndTime) | void |

**Applying TimeDrag in QTP**

The following code example illustrates how to reschedule the timeline of the appointment in the schedule control.

```
[For Schedule Control]
SwfWindow("GridSchedularDemo").SwfObject("Schedular").
TimeDrag(“Appointment1”, “ 10: 30:00:23””, “11:30:00:23”, “ 10:
30:00:23””, “11:30:00:23”)
```

## 7.6 Essential Diagram

### 7.6.1 How to change the node to a new position

**Supported method to change the node to a new position**

The MoveNode method is used to change the node to the new position.

**Use Case Scenarios**

This feature enables you to change the node to the new position of chart control in QTP testing.

**Methods Table**

| Method | Description | Parameters | Return Type |
|---|---|---|---|
| MoveNode | Changes the node to the new position. | public void MoveNode(string name, float offsetX, float offsetY) | void |

**Applying MoveNode in QTP**

The following code examples illustrate how to change the node to a new position in the chart control.

```
[For Diagram Control]
SwfWindow("Simple Flow Diagram").SwfObject("diagram1").SelectNode
“EllipseStart”
SwfWindow("Simple Flow Diagram").SwfObject("diagram1").MoveNode
“EllipseStart”,130.000000,35.000000
```

### 7.6.2 How to connect the specified nodes using connectors

**Supported method to connect the specified nodes using connectors**

The ConnectNodes method is used to connect the specified nodes using connectors

**Use Case Scenarios**

This feature enables you to connect the specified nodes using connectors in the chart control in QTP testing.

**Methods Table**

| Method | Description | Parameters | Return Type |
|---|---|---|---|
| ConnectNodes | Connects the specified nodes using connectors. | public void ConnectNodes(string startNode, string endNode, string connector) | void |

**Applying ConnectNodes in QTP**

The following code examples illustrate how to connect the specified nodes using connectors in the chart control.

```
[For Diagram Control]
SwfWindow("Simple Flow Diagram").SwfObject("diagram1").SelectNode
“EllipseStart”
SwfWindow("Simple Flow Diagram").SwfObject("diagram1").ConnectNodes
“RectangleProcess”, “RectangleProcess”, “LineConnector”
```

## 7.6.3 How to resize the node

**Supported method to resize the node in the diagram control**

The ResizeNode method is used to resize the node size in diagram control.

**Use Case Scenarios**

This feature enables you to resize the node in the diagram control.

**Methods Table**

| Method | Description | Parameters | Return Type |
|---|---|---|---|

| | | | |
|---|---|---|---|
| ResizeNode | Resizes the node in diagram control. | public void ResizeNode(string name, float offsetX, float offsetY) | void |

**Applying ResizeNode in QTP**

The following code examples illustrate how to resize the node size in the diagram control.

```
[For Diagram Control]
SwfWindow("Simple Flow Diagram").SwfObject("diagram1").SelectNode
“EllipseStart”
SwfWindow("Simple Flow Diagram").SwfObject("diagram1").ResizeNode
“LineConnecctor1”,-42.467853,0.000000
```

# Index

A

Assembly information 18

Automatic Configuration 14

C

Configuration 14

Configuring the SwfConfig file 67

Creating and Recording a Test 22

D

Documentation 6

E

Editing a Test 35

Essential Chart 61, 84

Essential Diagram 63, 88

Essential Grid 43, 65, 79

Essential Schedule 63, 86

Essential Tools 51, 65, 82

F

Frequently Asked Questions 71

G

General 65, 71

Getting Started 22

H

How do I know that Essential QuickTest Professional works as expected? 78

How to change the node to a new position 88

How to check and uncheck the CheckBoxAdv 82

How to collapse and expand the specified node in TreeViewADV 83

How to connect the specified nodes using connectors 88

How to fetch installation information related to the Syncfusion QTP add-on 76

How to find the count of a series within the chart 85

How to find the maximum Y-axis value in the specified region 85

How to get the description of the Check Box Cells and Normal Cells in Essential Grid 80

How to get the displayed text in the X-axis and Y-axis 84

How to know whether my swfconfig file holds an invalid assembly path reference 73

How to manually configure Syncfusion control to work with QTP 71

How to reschedule the appointment to another timeline 86

How to reschedule the timeline of an appointment 87

How to resize the node 89

How to select the XPtool bar without ID 82

How to set the current cell in Grid 81

I

Installation 7

Installation and Configuration 7

Installation and Deployment 7

Introduction to Essential QuickTest Professional 4

K

Known Issues 65

L

Licensing, Patches and Uninstallation 18

M

Manual Configuration 15

P

Prerequisites and Compatibility 5

R

Running a Test 33

Running the Saved Test 40

S

Sample and Location 17

Saving a Test 39

Supported Controls and Methods 43

**U**

Utilities 67

**W**

Why are Syncfusion controls not recognized even after adding the custom libraries? 78