

OPTIMIZE

MERCURY QUICKTEST PROFESSIONAL™

.NET ADD-IN

VERSION 9.2

Guide

MERCURY™

BUSINESS TECHNOLOGY OPTIMIZATION

Mercury QuickTest Professional .NET Add-in

Guide

Version 9.2

Document Release Date: February 26, 2007

MERCURY™

Mercury QuickTest Professional .NET Add-in Guide, Version 9.2

This document, and the accompanying software and other documentation, is protected by U.S. and international copyright laws, and may be used only in accordance with the accompanying license agreement. Features of the software, and of other products and services of Mercury Interactive Corporation, may be covered by one or more of the following patents: United States: 5,511,185; 5,657,438; 5,701,139; 5,870,559; 5,958,008; 5,974,572; 6,137,782; 6,138,157; 6,144,962; 6,205,122; 6,237,006; 6,341,310; 6,360,332; 6,449,739; 6,470,383; 6,477,483; 6,549,944; 6,560,564; 6,564,342; 6,587,969; 6,631,408; 6,631,411; 6,633,912; 6,694,288; 6,738,813; 6,738,933; 6,754,701; 6,792,460 and 6,810,494. Australia: 763468 and 762554. Other patents pending. All rights reserved.

U.S. GOVERNMENT RESTRICTED RIGHTS. This Software Documentation is a “commercial item” as defined at 48 C.F.R. 2.101 (October 1995). In accordance with 48 C.F.R. 12.212 (October 1995), 48 C.F.R. 27.401 through 27.404 and 52.227-14 (June 1987, as amended) and 48 C.F.R. 227.7201 through 227.7204 (June 1995), and any similar provisions in the supplements to Title 48 of the C.F.R. (the “Federal Acquisition Regulation”) of other entities of the U.S. Government, as applicable, all U.S. Government users acquire and may use this Documentation only in accordance with the restricted rights set forth in the license agreement applicable to the Computer Software to which this Documentation relates.

Mercury, Mercury Interactive, the Mercury logo, the Mercury Interactive logo, LoadRunner, WinRunner, SiteScope and TestDirector are trademarks of Mercury Interactive Corporation and may be registered in certain jurisdictions. The absence of a trademark from this list does not constitute a waiver of Mercury's intellectual property rights concerning that trademark.

All other company, brand and product names may be trademarks or registered trademarks of their respective holders. Mercury disclaims any responsibility for specifying which marks are owned by which companies or which organizations.

Mercury provides links to external third-party Web sites to help you find supplemental information. Site content and availability may change without notice. Mercury makes no representations or warranties whatsoever as to site content or availability.

Mercury Interactive Corporation
379 North Whisman Road
Mountain View, CA 94043
Tel: (650) 603-5200
Fax: (650) 603-5300
<http://www.mercury.com>

© 1992 - 2007 Mercury Interactive Corporation, All rights reserved

If you have any comments or suggestions regarding this document, please send them by e-mail to documentation@mercury.com.

Table of Contents

| | |
|---|----------|
| Welcome to This Guide | 5 |
| How This Guide Is Organized | 6 |
| Who Should Read This Guide | 7 |
| QuickTest Professional Online Documentation | 8 |
| Additional Online Resources..... | 10 |
| Documentation Updates | 11 |
| Typographical Conventions..... | 12 |

PART I: SETTING UP .NET ADD-IN SUPPORT

| | |
|--|-----------|
| Chapter 1: Installing the .NET Add-in | 15 |
| Before You Install | 15 |
| Running the Setup Program | 16 |
| Chapter 2: Activating .NET Support | 25 |
| Loading QuickTest with .NET Add-in Support | 26 |
| Loading QuickTest without .NET Add-in Support..... | 28 |
| Repairing and Uninstalling the .NET Add-in..... | 29 |

PART II: WORKING WITH THE .NET ADD-IN

| | |
|---|-----------|
| Chapter 3: Testing .NET Web Forms Applications | 35 |
| About Testing .NET Web Forms Objects | 35 |
| Recording Tests or Components on .NET Web Forms Objects | 37 |
| Checking .NET Web Forms Objects and Outputting Values | 40 |
| Using .NET Web Forms Objects, Methods, and Properties to Enhance Your Test or Component..... | 41 |
| Running Tests or Components on .NET Web Forms Objects..... | 42 |

| | |
|---|-----------|
| Chapter 4: Testing .NET Windows Forms Applications..... | 43 |
| About Testing .NET Windows Forms Objects | 44 |
| Recording Tests or Components on .NET Windows Forms Objects .. | 45 |
| Checking .NET Windows Forms Objects and Outputting Values | 48 |
| Using .NET Windows Forms Objects, Methods, and Properties to Enhance Your Test or Component..... | 50 |
| Running Tests or Components on .NET Windows Forms Objects..... | 51 |
| Using the .NET Windows Forms Spy | 52 |
| Chapter 5: Testing Windows Presentation Foundation Applications..... | 65 |
| About Testing WPF Objects..... | 66 |
| About WPF User Interface Automation | 67 |
| Recording Tests or Components on WPF Objects | 68 |
| Checking WPF Objects and Outputting Values..... | 71 |
| Using WPF Objects, Methods, and Properties to Enhance Your Test or Component | 72 |
| Running Tests or Components on WPF Objects | 74 |
| Index..... | 75 |

Welcome to This Guide

Welcome to the QuickTest Professional .NET Add-in.

The QuickTest Professional .NET Add-in recognizes and records on objects based on .NET Windows Forms libraries, Web Forms controls, or Windows Presentation Foundation controls, and accesses properties and methods supported by those controls.

Note: QuickTest Professional .NET Add-in Extensibility enables you to support testing on third-party and custom .NET controls that are not supported out-of-the-box by the QuickTest Professional .NET Add-in. For more information, refer to the *QuickTest Professional .NET Add-in Extensibility Developer's Guide*.

| This chapter describes: | On page: |
|---|----------|
| How This Guide Is Organized | 6 |
| Who Should Read This Guide | 7 |
| QuickTest Professional Online Documentation | 8 |
| Additional Online Resources | 10 |
| Documentation Updates | 11 |
| Typographical Conventions | 12 |

How This Guide Is Organized

This guide explains how to install the QuickTest Professional .NET Add-in and use QuickTest Professional to test .NET Windows Forms, Web Forms, and Windows Presentation Foundation applications.

This guide should be used in conjunction with the *QuickTest Professional User's Guide* and the *QuickTest Professional Object Model Reference*. In addition, the *QuickTest Professional .NET Add-in Extensibility Developer's Guide* explains how to create support for custom .NET controls. These guides can be accessed online by choosing **Help > QuickTest Professional Help** from the QuickTest main window.

This guide contains:

Part I Setting Up .NET Add-in Support

Details the process of installing the QuickTest Professional .NET Add-in, including:

- ▶ Installing the .NET Add-in
- ▶ Activating .NET Support

Part II Working with the .NET Add-in

Explains how to use the QuickTest Professional .NET Add-in to test .NET Web Forms objects, including:

- ▶ Testing .NET Web Forms Applications
- ▶ Testing .NET Windows Forms Applications
- ▶ Testing Windows Presentation Foundation Applications

Note: The information, examples, and screen captures in this guide focus specifically on working with QuickTest tests. However, much of the information applies equally to components.

Business components and scripted components are part of Mercury Business Process Testing, which utilizes a keyword-driven methodology for testing applications. For more information, refer to the *QuickTest Professional User's Guide* and the *QuickTest Professional for Business Process Testing User's Guide*.

Who Should Read This Guide

This guide is intended for QuickTest Professional users at all levels who want to use the QuickTest Professional .NET Add-in to test .NET Windows Forms, Web Forms, and Windows Presentation Foundation applications.

Readers should already have some understanding of functional testing concepts and processes, and know which aspects of their application they want to test.

Note: The .NET Add-in takes advantage of commonly used QuickTest features such as the object repository, Keyword View, and checkpoints and output value steps to enable you to test your application. You should have at least a basic understanding of these concepts before you begin working with the QuickTest Professional .NET Add-in.

QuickTest Professional Online Documentation

QuickTest Professional includes the following online documentation:

Readme provides the latest news and information about QuickTest. Choose **Start > Programs > QuickTest Professional > Readme**.

QuickTest Professional Installation Guide explains how to install and set up QuickTest. Choose **Help > Printer-Friendly Documentation > Mercury QuickTest Professional Installation Guide**.

QuickTest Professional Tutorial teaches you basic QuickTest skills and shows you how to design tests for your applications. Choose **Help > QuickTest Professional Tutorial**.

Product Feature Movies provide an overview and step-by-step instructions describing how to use selected QuickTest features. Choose **Help > Product Feature Movies**.

Printer-Friendly Documentation displays the complete documentation set in Adobe portable document format (PDF). Online books can be viewed and printed using Adobe Reader, which can be downloaded from the Adobe Web site (<http://www.adobe.com>). Choose **Help > Printer-Friendly Documentation**.

QuickTest Professional Help includes:

- **What's New in QuickTest Professional** describes the newest features, enhancements, and supported environments in the latest version of QuickTest.
- **QuickTest User's Guide** describes how to use QuickTest to test your application.
- **QuickTest for Business Process Testing User's Guide** provides step-by-step instructions for using QuickTest to create and manage assets for use with Business Process Testing.
- **QuickTest Object Model** describes QuickTest test objects, lists the methods and properties associated with each object, and provides syntax information and examples for each method and property.

- ▶ **QuickTest Advanced References** contains documentation for the following QuickTest COM and XML references:
 - ▶ **QuickTest Automation** provides syntax, descriptive information, and examples for the automation objects, methods, and properties. It also contains a detailed overview to help you get started writing QuickTest automation scripts. The automation object model assists you in automating test management, by providing objects, methods and properties that enable you to control virtually every QuickTest feature and capability.
 - ▶ **QuickTest Test Results Schema** documents the XML schema that enables you to customize your test results.
 - ▶ **QuickTest Test Object Schema** documents the XML schema that enables you to extend test object support in different environments.
 - ▶ **QuickTest Object Repository Automation** documents the Object Repository automation object model that enables you to manipulate QuickTest object repositories and their contents from outside of QuickTest.
- ▶ **VBScript Reference** contains Microsoft VBScript documentation, including VBScript, Script Runtime, and Windows Script Host.

To access the QuickTest Professional Help, choose **Help > QuickTest Professional Help**. You can also access the QuickTest Professional Help by clicking in selected QuickTest windows and dialog boxes and pressing F1. Additionally, you can view a description, syntax, and examples for a QuickTest test object, method, or property by placing the cursor on it and pressing F1.

Note: Your QuickTest Help may contain additional items relevant to any QuickTest add-ins you have installed. For more information, refer to the relevant add-in documentation.

Additional Online Resources

Knowledge Base uses your default Web browser to open the Mercury Customer Support Web site directly to the Knowledge Base landing page. Choose **Help > Knowledge Base**. The URL for this Web site is <http://support.mercury.com/cgi-bin/portal/CSO/kbBrowse.jsp>.

Customer Support Web Site uses your default Web browser to open the Mercury Customer Support Web site. This site enables you to browse the Mercury Support Knowledge Base and add your own articles. You can also post to and search user discussion forums, submit support requests, download patches and updated documentation, and more. Choose **Help > Customer Support Web Site**. The URL for this Web site is <http://support.mercury.com>.

Send Feedback enables you to send online feedback about **QuickTest Professional** to the product team. Choose **Help > Send Feedback**.

Mercury Home Page uses your default Web browser to access the HP Software Web site. This site provides you with the most up-to-date information on HP Software products. This includes new software releases, seminars and trade shows, customer support, and more. Choose **Help > Mercury Home Page**.

Mercury Best Practices contain guidelines for planning, creating, deploying, and managing a world-class IT environment. Mercury provides three types of best practices: Process Best Practices, Product Best Practices, and People Best Practices. Licensed customers of Mercury software can read and use the Mercury Best Practices available from the Customer Support site, <http://support.mercury.com>.

Documentation Updates

Mercury is continually updating its product documentation with new information. You can download the latest version of this document from the Customer Support Web site (<http://support.mercury.com>).

To download updated documentation:

- 1** If you are not already logged into the Customer Support Web site, enter your login information and click **Login**.
- 2** Click the **Documentation** link.
- 3** Under **Please Select Product**, select **QuickTest Professional**.

Note that if the required product does not appear in the list, you must add it to your customer profile. Click **My Account** to update your profile.

- 4** Click **Retrieve**. The Documentation page opens and lists the documentation available for the current release and for previous releases. If a document was updated recently, **Updated** appears next to the document name.
- 5** Click a document link to download the documentation.

Typographical Conventions

This guide uses the following typographical conventions:

| | |
|---------------------------------------|---|
| UI Elements and Function Names | This style indicates the names of interface elements on which you perform actions, file names or paths, and other items that require emphasis. For example, “Click the Save button.” It also indicates method or function names. For example, “The wait_window statement has the following parameters:” |
| <i>Arguments</i> | This style indicates method, property, or function arguments and book titles. For example, “Refer to the <i>Mercury User’s Guide</i> .” |
| <Replace Value> | Angle brackets enclose a part of a file path or URL address that should be replaced with an actual value. For example, <MyProduct installation folder>\bin . |
| Example | This style is used for examples and text that is to be typed literally. For example, “Type Hello in the edit box.” |
| CTRL+C | This style indicates keyboard keys. For example, “Press ENTER.” |
| [] | Square brackets enclose optional arguments. |
| { } | Curly brackets indicate that one of the enclosed values must be assigned to the current argument. |
| ... | In a line of syntax, an ellipsis indicates that more items of the same format may be included. In a programming example, an ellipsis is used to indicate lines of a program that were intentionally omitted. |
| | A vertical bar indicates that one of the options separated by the bar should be selected. |

Part I

Setting Up .NET Add-in Support

1

Installing the .NET Add-in

This chapter lists the pre-installation requirements and explains how to install the .NET Add-in.

| This chapter describes: | On page: |
|---------------------------|----------|
| Before You Install | 15 |
| Running the Setup Program | 16 |

Before You Install

Before you begin to install the .NET Add-in, review the system requirements listed below.

To work successfully with the .NET Add-in, your system configuration should meet the requirements as specified for QuickTest Professional 9.2 (in the *QuickTest Professional 9.2 Readme*), plus the following add-in specific requirements:

Prerequisites: QuickTest Professional 9.2, including the Web Add-in.

Hard Disk Space: 71 MB of free hard disk space is required on the disk on which you want to install the .NET Add-in, in addition to the space required for the QuickTest Professional installation.

In addition, the disk on which Windows is installed (system disk) requires 98 MB of free hard disk space while the .NET Add-in is being installed. After the installation is complete, 70 MB of this disk space is released (meaning that 28 MB of the system disk remains in use by QuickTest Professional).

Additionally, to work with objects based on WPF (Windows Presentation Foundation) controls, your system should meet all the requirements necessary to run WPF applications.

Running the Setup Program

The Setup program installs add-in support for testing .NET Windows Forms, Web Forms, and Windows Presentation Foundation applications in your QuickTest Professional installation folder.

Notes:

- ▶ To install the .NET Add-in, you must be logged on with administrator privileges.
 - ▶ You must not run any other installation at the same time as you run the .NET Add-in installation.
-

To install the .NET Add-in:

- 1** Close any instances of QuickTest Professional. It is also recommended to close all other open applications.
- 2** Insert the CD-ROM into the CD-ROM drive.
 - ▶ If the CD-ROM drive is on your local computer, the QuickTest Professional .NET Add-in Setup window opens.

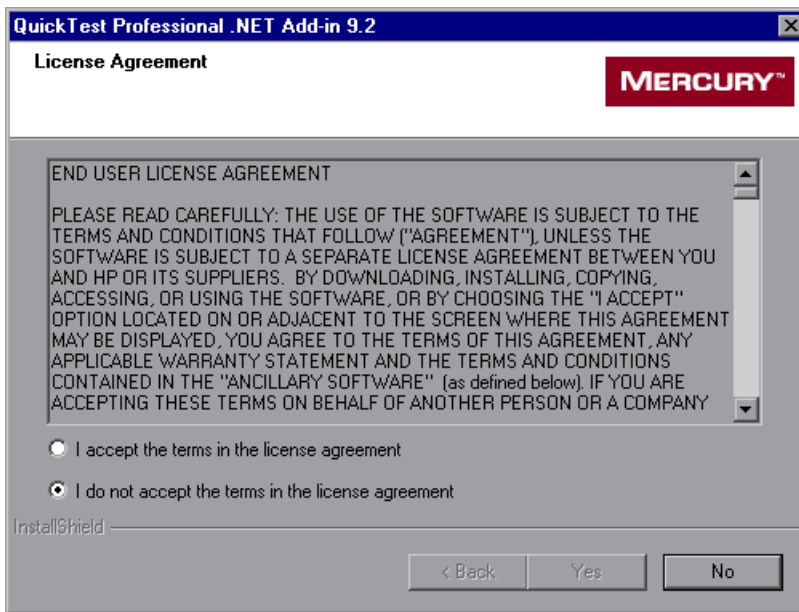
- ▶ If you are installing from a network drive, browse to it and double-click **autorun.exe** in the root folder of the CD-ROM. The QuickTest Professional .NET Add-in Setup window opens.



The QuickTest Professional .NET Add-in Setup window contains the following options:

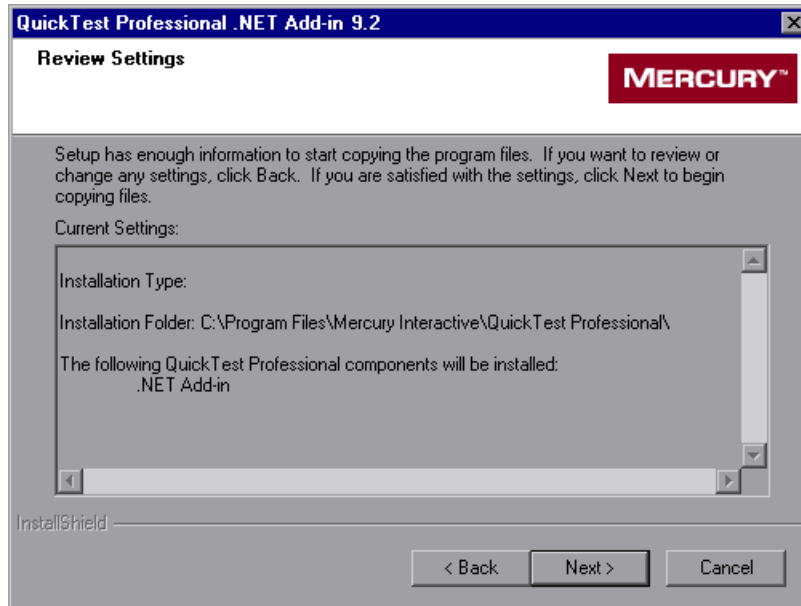
- ▶ **Add-in Setup.** Starts the .NET Add-in installation program.
- ▶ **Product Information.** Opens the product information site (<http://www.mercury.com/us/products/quality-center/functional-testing/quicktest-professional/>).
- ▶ **Readme.** Opens the *QuickTest Professional .NET Add-in Readme* file.
- ▶ **Contact Mercury.** Opens the contact information page on the Mercury Web site (<http://www.mercury.com/us/company/corporate-info/contact-us/>).

- ▶ **Support.** Opens the Mercury Customer Support Web site (<http://support.mercury.com>).
 - ▶ **Browse CD.** Displays the contents of the QuickTest Professional .NET Add-in CD-ROM.
 - ▶ **Exit.** Exits the Setup program.
- 3 To start the .NET Add-in Setup program, click **Add-in Setup**. The .NET Add-in Setup program starts.
 - 4 The License Agreement screen opens. Read the agreement.



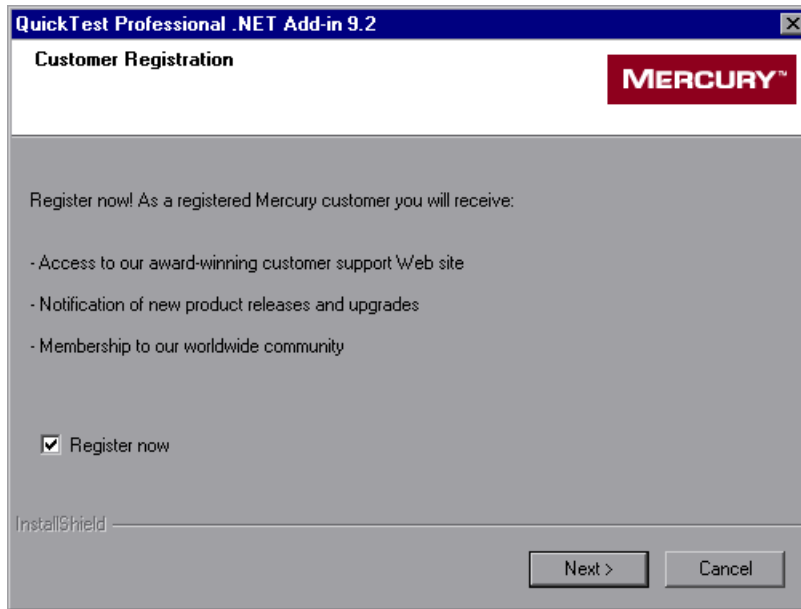
To install QuickTest Professional add-in support for testing .NET applications, you must accept the terms of the license agreement by selecting the **I accept the terms in the license agreement** option and clicking **Yes**. If you click **No**, the Setup program closes.

- 5 In the Review Settings screen, review the installation settings.



To change your settings, click **Back**. To confirm the settings, click **Next**. The installation process begins.

- 6 In the Customer Registration screen, you can specify whether to register your copy of the QuickTest Professional .NET Add-in.



If you register, you receive:

- ▶ access to the Mercury award-winning Customer Support Web site.
- ▶ notification of new product releases and upgrades.
- ▶ membership to the Mercury worldwide community of testers.

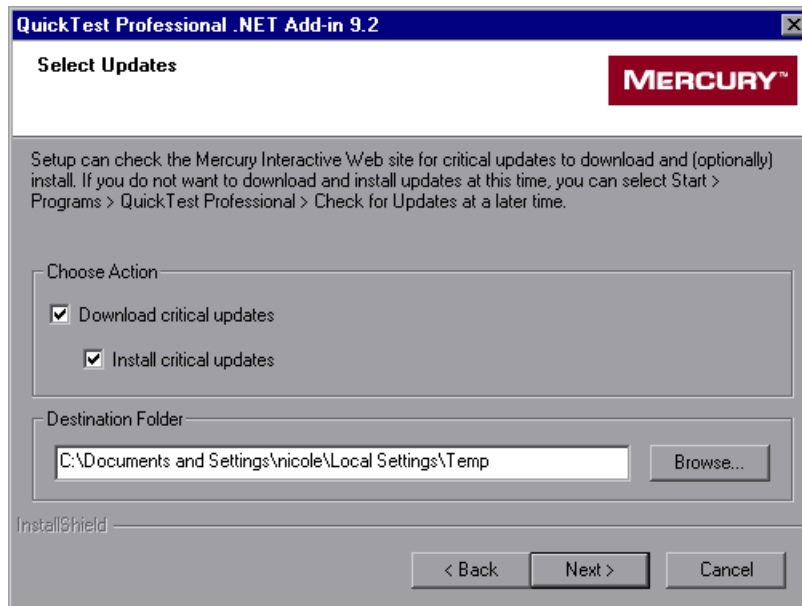
To register now, confirm that the **Register now** check box is selected. Click **Next**. Your browser opens to the Mercury Customer Support site (<http://support.mercury.com>). Follow the directions on the Customer Support site.

To register at a later time, or if you are already a registered user, clear the **Register now** check box and click **Next** to proceed with the installation.

- 7 The installation process checks whether there are any critical updates to be installed for the version of the .NET Add-in you are installing. It also checks whether there are any critical updates to be installed for your version of QuickTest Professional and any other external add-ins that are installed on your computer.

If no critical updates are found, the installation continues with step 9.

If the Setup program finds any critical updates to be installed, the Select Updates screen opens.



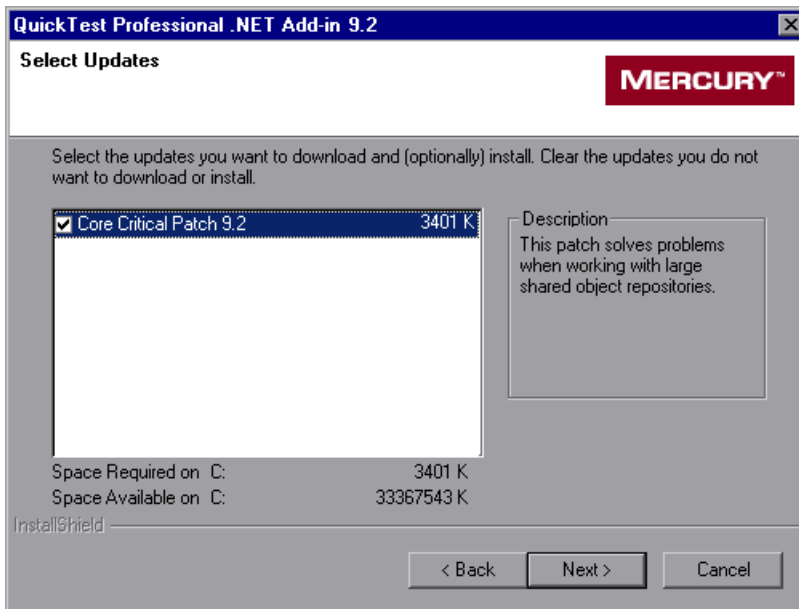
Choose one of the following options:

- **Download critical updates.** Downloads any critical updates to the specified destination folder. This option is selected by default.
- **Install critical updates.** Installs any downloaded critical updates. This option is selected by default.

Note: If the Setup program finds any critical updates to be installed, it is highly recommended to install them immediately. (You can also check for updates at a later time by choosing **Start > Programs > QuickTest Professional > Check for Updates.**)

A default download folder is displayed in the **Destination Folder** box. To select a different location to which to download the file, click **Browse**, choose a folder, and click **OK**.

Click **Next** to proceed. If you chose to download and (optionally) install critical updates, a list of all available critical updates opens. Otherwise, continue with step 9.



- 8 Select the updates that you want to download and (optionally) install. Clear the updates that you do not want to download or install.

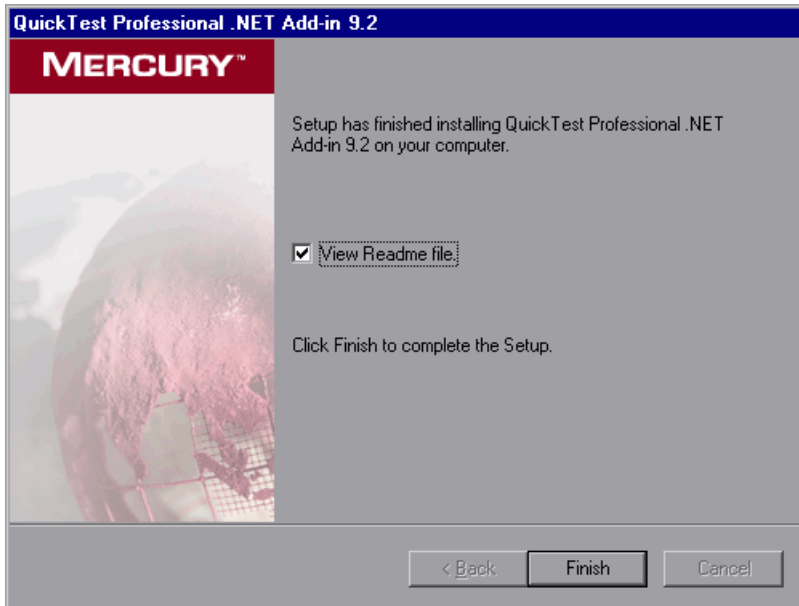
Click **Next** to proceed. QuickTest downloads the specified critical updates, according to the option you selected in step 7. If you chose to download and install the updates, they are installed at the end of the add-in installation process, after you restart your computer. (Note that you are always prompted to restart your computer if you choose to download and install critical updates.)

- 9 If the screen prompts you to restart your computer, you can choose to restart your computer at a later time, but you must restart your computer before you use QuickTest Professional.

Note: If you are prompted to restart, it is strongly recommended that you do so as soon as possible. Delaying the system restart could result in unexpected system behavior.

If you are not prompted to restart, proceed to step 10. Otherwise, step 10 occurs after the restart.

- 10 The InstallShield Wizard Complete screen opens. If you want to open the *QuickTest Professional .NET Add-in Readme* file at the end of the Setup process, select the **View Readme file** check box.



Click **Finish** to complete the Setup program.

In the Readme file, you can view the latest technical and troubleshooting information for the .NET Add-in. To open the Readme file at another time, choose **Start > Programs > QuickTest Professional > Add-ins > .NET Add-in Readme**.

2

Activating .NET Support

Before you can work with the QuickTest Professional .NET Add-in, you must make sure that the appropriate .NET Add-in items are loaded. You can load QuickTest without add-in support for .NET if you do not want to test .NET applications. You can also repair or uninstall your .NET installation. Your QuickTest Professional license enables all QuickTest features including the use of all QuickTest add-ins.

| This chapter describes: | On page: |
|---|-----------------|
| Loading QuickTest with .NET Add-in Support | 26 |
| Loading QuickTest without .NET Add-in Support | 28 |
| Repairing and Uninstalling the .NET Add-in | 29 |

Loading QuickTest with .NET Add-in Support

You use the Add-in Manager to load support for testing .NET applications.

Notes:

- ▶ For optimal performance when testing .NET Add-in applications, it is strongly recommended that you work with *only* the required add-in items loaded. Note that the QuickTest Professional Web Add-in does not always need to be loaded when using the .NET Add-in; it is required only if you are testing .NET Web Forms.
 - ▶ When testing applications that do not contain .NET objects, it is strongly recommended that you do not load the .NET or WPF items.
-

To start QuickTest with add-in support for testing .NET applications:

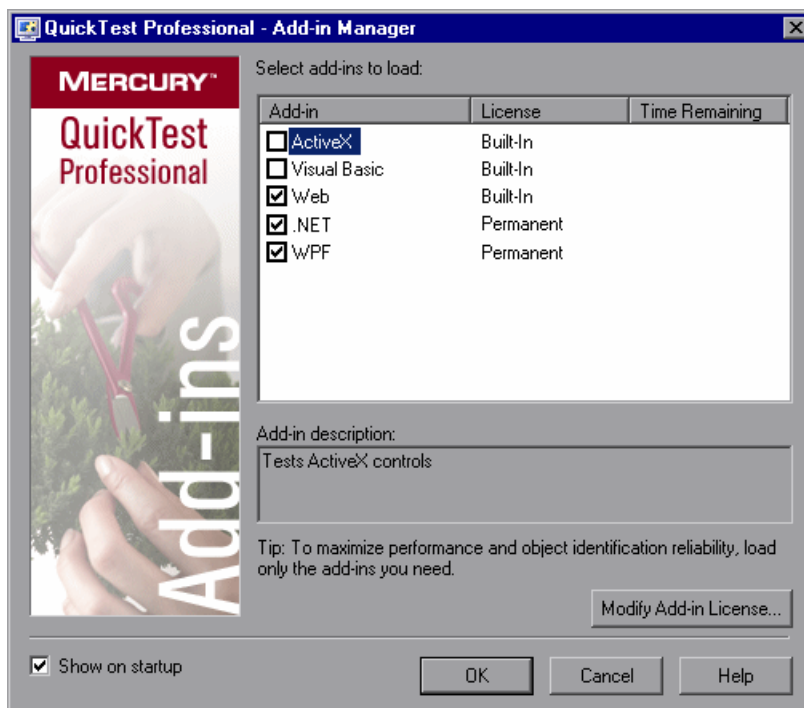
- 1** Choose **Start > Programs > QuickTest Professional > QuickTest Professional**. The QuickTest Professional Add-in Manager dialog box opens.

(If the Add-in Manager dialog box does not open, see “Displaying the Add-in Manager Dialog Box” on page 28.)

- 2** If you want to test .NET Windows Forms, select **.NET**. When you click **OK** (in step 3), a message is displayed stating that for full operation of the .NET Add-in you must also load the Web Add-in. If you want to test only .NET Windows Forms (and not .NET Web Forms), click **Yes**.

If you want to test .NET Web Forms, select both **.NET** and **Web**.

If you want to test Windows Presentation Foundation objects, select **WPF**. If you want to test only WPF controls, it is not necessary to select the **.NET** item.



3 Click **OK**.

For more information on the Add-in Manager dialog box, refer to the *QuickTest Professional User's Guide*.

Displaying the Add-in Manager Dialog Box

You can set an option in QuickTest to determine whether the Add-in Manager opens when you open QuickTest, or whether it automatically loads the same add-ins that were loaded in the previous QuickTest session.

To instruct the Add-in Manager dialog box to open when you open QuickTest:

- 1** Choose **Start > Programs > QuickTest Professional > QuickTest Professional** to start QuickTest.
- 2** From the QuickTest menu, choose **Tools > Options** and click the **General** tab.
- 3** Select **Display Add-in Manager on startup**.
- 4** Click **OK** to close the Options dialog box.
- 5** Close and reopen QuickTest.

Loading QuickTest without .NET Add-in Support

If you want to work with QuickTest without support for .NET, you can load QuickTest without the .NET Add-in.

To load QuickTest without add-in support for .NET:

- 1** Choose **Start > Programs > QuickTest Professional > QuickTest Professional**. The QuickTest Professional Add-in Manager dialog box opens. (If the Add-in Manager dialog box does not open, see “Displaying the Add-in Manager Dialog Box” above.)
- 2** Clear the **.NET** and **WPF** check boxes and click **OK**. QuickTest opens without add-in support for .NET.

Repairing and Uninstalling the .NET Add-in

You can repair a QuickTest Professional .NET Add-in installation that has become corrupted. In addition, you can uninstall the QuickTest Professional .NET Add-in without uninstalling QuickTest or any other add-ins.

Note: You can also uninstall the QuickTest Professional application entirely. If you uninstall QuickTest Professional, the uninstall program removes all installed features, including any external add-ins that are installed. For more information on uninstalling QuickTest Professional, refer to the *QuickTest Professional Installation Guide*.

Repairing the .NET Add-in Installation

Your QuickTest Professional .NET Add-in 9.2 CD-ROM enables you to repair an existing .NET Add-in 9.2 installation by replacing any missing or damaged files in your current .NET Add-in installation.

To repair your .NET Add-in installation:

- 1** Insert the QuickTest Professional .NET Add-in CD-ROM into the CD-ROM drive. If the CD-ROM drive is on your local computer, the QuickTest Professional .NET Add-in Setup window opens.

If the CD-ROM is in a network drive, double-click **autorun.exe** in the root folder of the CD-ROM. The QuickTest Professional .NET Add-in Setup window opens.

Note: You must use the CD-ROM with the exact same version of the .NET Add-in that you used for the original installation.

- 2** Click **Add-in Setup**. The Welcome screen opens.

- 3 In the Welcome screen, select **Repair** and click **Next**.

The maintenance program repairs your QuickTest Professional .NET Add-in installation.

- 4 The Maintenance Complete screen may prompt you to restart your computer. If it does, you can choose to restart your computer at a later time, but you must restart your computer before you use QuickTest Professional.

Note: You can save any open files, but you should restart your computer as soon as possible. Delaying the system restart could result in unexpected system behavior.

- 5 Click **Finish** to complete the repair process.

Uninstalling the .NET Add-in

You can uninstall the .NET Add-in using either the **Add/Remove Programs** option in the Windows Control Panel, or using the QuickTest Professional .NET Add-in CD-ROM.

To uninstall the .NET Add-in using the Add/Remove Programs option:

- 1 Make sure that QuickTest is closed.
- 2 Choose **Start > Settings > Control Panel**.
- 3 Double-click the **Add/Remove Programs** option.
- 4 In the Add/Remove Programs dialog box, select **QuickTest Professional .NET Add-in** and then click **Change/Remove**.
- 5 A message prompts you to confirm your decision to uninstall the QuickTest Professional .NET Add-in. Click **Yes** to uninstall the .NET Add-in. The uninstall program removes the .NET Add-in from your computer. QuickTest Professional and any other installed add-ins remain on your computer.

Note: Clicking **No** keeps the .NET Add-in installed on your computer.

- 6 The Maintenance Complete screen may prompt you to restart your computer. If it does, you can choose to restart your computer at a later time, but you must restart your computer before you use QuickTest Professional.
-

Note: You can save any open files, but you should restart your computer as soon as possible. Delaying the system restart could result in unexpected system behavior.

- 7 Click **Finish** to complete the uninstall process.

To uninstall the .NET Add-in using the QuickTest Professional .NET Add-in CD-ROM:

- 1 Insert the QuickTest Professional .NET Add-in CD-ROM into the CD-ROM drive. If the CD-ROM drive is on your local computer, the QuickTest Professional .NET Add-in Setup window opens.

If the CD-ROM is in a network drive, double-click **autorun.exe** in the root folder of the CD-ROM. The QuickTest Professional .NET Add-in Setup window opens.

Note: You must use the CD-ROM with the exact same version of the .NET Add-in that you used for the original installation.

- 2 Click **Add-in Setup**. The Welcome screen opens.
- 3 In the Welcome screen, select **Remove** and click **Next**.
- 4 A message prompts you to confirm your decision to uninstall the QuickTest Professional .NET Add-in. Click **OK** to uninstall the .NET Add-in. The uninstall program removes the .NET Add-in from your computer. QuickTest Professional and any other installed add-ins remain on your computer.

Note: Clicking **Cancel** keeps the .NET Add-in installed on your computer and returns to the previous screen.

- 5** The Maintenance Complete screen may prompt you to restart your computer. If it does, you can choose to restart your computer at a later time, but you must restart your computer before you use QuickTest Professional.

Note: You can save any open files, but you should restart your computer as soon as possible. Delaying the system restart could result in unexpected system behavior.

- 6** Click **Finish** to complete the uninstall process.

Part II

Working with the .NET Add-in

3

Testing .NET Web Forms Applications

You can use the .NET Add-in to test .NET Web Forms controls. You can also create checkpoints and output values on .NET Web Forms objects to verify that the objects in your application function as expected.

| This chapter describes: | On page: |
|---|----------|
| About Testing .NET Web Forms Objects | 35 |
| Recording Tests or Components on .NET Web Forms Objects | 37 |
| Checking .NET Web Forms Objects and Outputting Values | 40 |
| Using .NET Web Forms Objects, Methods, and Properties to Enhance Your Test or Component | 41 |
| Running Tests or Components on .NET Web Forms Objects | 42 |

About Testing .NET Web Forms Objects

You can use the .NET Add-in to test most .NET Web Forms controls. You create and run steps on .NET Web Forms applications similar to the way you do with other Web applications.

Before or while creating tests, you can learn objects by adding them to the repository (using the **Add Objects** or **Navigate and Learn** options), or by recording on your application (which also adds the recorded objects to the object repository.) For more information on learning objects, refer to the *QuickTest Professional User's Guide*.

Note: When QuickTest learns .NET Web Forms objects, it does not learn the HTML elements that make up the test objects. For example, when QuickTest learns the WbfGrid test object, the WbfGrid object is the bottom-most object in the hierarchy, and the HTML elements used to create the grid's cells are not learned. If you want to learn an object that is automatically filtered out, you can do so by selecting it in the Object Selection dialog box.

After you have added .NET Web Forms objects to the object repository, you can use the Keyword View and Expert View to activate .NET Web Forms test object methods and run-time object methods, retrieve and set the values of properties, and check that objects exist.

You check the properties of a .NET Web Forms object in the same way as you check the properties of corresponding Web objects. When you create a checkpoint on a .NET Web Forms object, QuickTest stores the selected property values of the object. If your application changes, you can modify the captured values to match the new expected values.

When you run a test or component, QuickTest verifies the behavior of your application by comparing values stored with the checkpoint, to actual property values in your application. If the values do not match, the checkpoint fails. The results of the checkpoint can be viewed in the QuickTest Test Results window. For additional information on the Test Results window, refer to the *QuickTest Professional User's Guide*.

Note that this chapter provides examples using both the Keyword View and the Expert View. Some of the information provided and procedures described require working in the Expert View. For information on working in the Expert View, refer to the *QuickTest Professional User's Guide*.

Note: When you load the .NET Add-in, the settings in the Web Event Recording Configuration dialog box (**Tools > Web Event Recording Configuration**) are automatically customized to optimally record and run tests and components on .NET Web Forms applications. You do not need to make any Web event configuration changes. For this reason, it is recommended not to record and run tests or components on Web applications while the .NET Add-in is loaded.

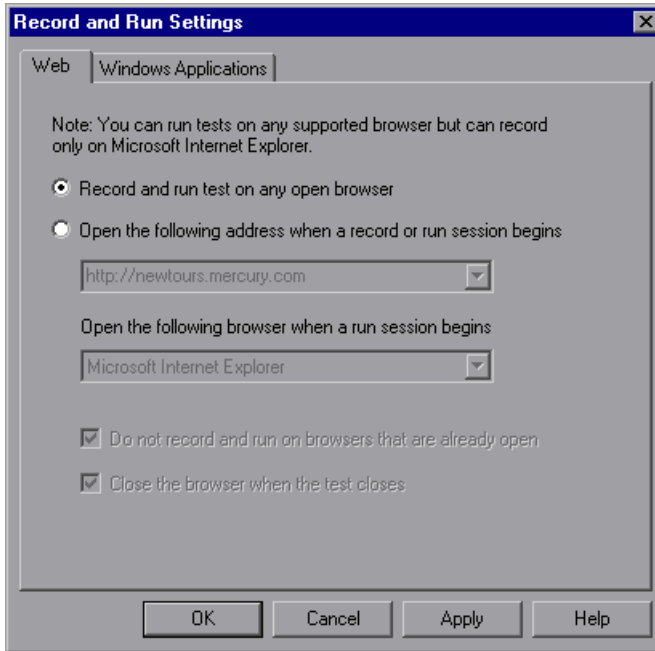
Recording Tests or Components on .NET Web Forms Objects

You record steps on .NET Web Forms objects similar to the way you record on corresponding Web objects. For more information, refer to the *QuickTest Professional User's Guide*.

The Record and Run Settings dialog box opens automatically each time you begin recording a new test (unless you open the dialog box and set your preferences manually before you begin recording). You can also open it by selecting **Automation > Record and Run Settings**.

Note: The Record and Run Settings dialog box applies only to tests. Record settings for components are specified in the Applications tab or Applications dialog box of the relevant application area. However, specific record and run settings do not need to be defined for components. For more information on the Applications tab and Applications dialog box, refer to the *QuickTest Professional for Business Process Testing User's Guide*.

You can use the Web tab of the Record and Run Settings dialog box to instruct QuickTest to open your .NET Web Forms application each time you begin a recording session, or you can instruct QuickTest to record on any open browser.



The Web Add-in must be loaded to view the Web tab of the Record and Run Settings dialog box. For information on loading add-ins, see “Activating .NET Support” on page 25.

For more information on the Record and Run Settings dialog box, refer to the *QuickTest Professional User's Guide*.

In addition to setting the appropriate settings in the Web tab, you should confirm that the other tabs in the dialog box have the appropriate settings.

The following settings are recommended:

- **Windows Applications tab.** Choose **Record and run only on:** and confirm that all three check boxes are cleared.
- **Other tabs.** (If displayed.) Choose the option to record and run on any open application (upper radio button of each tab).

While these settings do not directly affect your record or run sessions when working with .NET Web Forms applications, they prevent you from inadvertently recording operations performed on Windows applications (such as e-mail) during your recording session. These settings also prevent QuickTest from opening unnecessary applications when you record or run tests or components on .NET Web Forms applications.

When you record tests or components on a .NET Web Forms application, QuickTest recognizes .NET Web Forms objects. QuickTest inserts a step with the relevant .NET icon in the Keyword View and generates .NET-specific statements in the Expert View.

For example, if you record a click on a node in a .NET Web Forms tree, the Keyword View may be displayed as follows:

| Item | Operation | Value | Documentation |
|----------------|-----------|-------------|--|
| ▼ Action1 | | | |
| ▼ WebControls2 | | | |
| ▼ WebControls2 | | | |
| ▼ TreeView2 | Select | "Month;Mar" | Select the "Month;Mar" node in the "TreeView2" tree. |

QuickTest records these steps in the Expert View as:

```
Browser("WebControls2").Page("WebControls2").WbfTreeView("TreeView2").
    Select "Month;Mar"
```

For more information on .NET Web Forms test objects, methods, and properties, refer to the .NET Web Forms section of the *QuickTest Professional Object Model Reference*. For more information on creating tests and components, refer to the *QuickTest Professional User's Guide*.

Checking .NET Web Forms Objects and Outputting Values

You use checkpoints to check the properties of .NET Web Forms objects the same way you check the properties of Web objects. You can also output property or text values from the objects in your .NET Web Forms application to use in your test or component.

The following checkpoints and output values are supported when testing .NET Web Forms objects:

| Type | Checkpoint | Output Value |
|----------------------|------------|--------------|
| Standard | Yes | Yes |
| Page | Yes | Yes |
| Accessibility | Yes | No |
| Text | Yes | Yes |
| Table | Yes | Yes |
| Bitmap | Yes | No |
| XML (Web Page/Frame) | Yes | No |

You can check or output any test object property associated with an object. In general, each .NET Web Forms test object has the same properties as the corresponding Web test object, plus several .NET-specific properties. For a list and description of the test object properties associated with each .NET Web Forms object, refer to the .NET Web Forms section of the *QuickTest Professional Object Model Reference*.

To check properties that are not included in the Checkpoint Properties dialog box, use the **Object** property. For more information, see “Accessing Internal Properties and Methods of Run-Time .NET Web Forms Objects” on page 41.

For more information on checkpoints and output values, refer to the *QuickTest Professional User's Guide*.

Using .NET Web Forms Objects, Methods, and Properties to Enhance Your Test or Component

A test or component consists of statements coded in Microsoft VBScript. These statements are composed of objects, methods, and/or properties that instruct QuickTest to perform operations or retrieve information. You add these statements using objects from your object repositories, and methods and properties that are available for each object type. In addition, when you record, these statements are generated automatically in response to input to the application. You can also program statements manually, or mix recorded and programmed statements in the same test or component. You create, view, and edit these statements in the Keyword View and/or Expert View.

Working with Test Object Methods

QuickTest provides a variety of test object methods that you can use with .NET Web Forms objects. You can record some of these methods while recording on .NET Web Forms objects. You can add additional functionality to your test or component by entering statements manually in the Keyword View or Expert View.

The .NET Add-in supports IntelliSense and statement completion in the Expert View, including the display of run-time (native) methods and properties when using the **Object** property in a statement. The .NET Add-in also supports generating statements that access run-time methods and properties in the Step Generator.

For more information, refer to the *QuickTest Professional User's Guide*.

Accessing Internal Properties and Methods of Run-Time .NET Web Forms Objects

You can use the **Object** property to retrieve internal (native) properties and activate internal methods of any .NET Web Forms object in your application.

In the example below, the **orientation** property of the **WbfTabStrip** control is returned and displayed in a message box.

```
MsgBox Browser("WebControls:").Page("Page").WbfTabStrip("WbfTabStrip").  
    Object.Orientation
```

The **Object** property is also useful for verifying the value of properties that are not available using a standard checkpoint.

For additional information on the **Object** property and for information on .Net Web Forms test objects and methods, refer to the .NET Web Forms section of the *QuickTest Professional Object Model Reference*.

Running Tests or Components on .NET Web Forms Objects

You run tests or components containing .NET Web Forms objects in the same way as you run any other test or component.

The test results tree displays the same .NET Web Forms icons as those used in the Keyword View. If you choose to save still images or movies to the test results (**Tools > Options > Run** tab), the bottom right pane of the Test Results window displays the page that was captured during the run session and highlights the .NET Web Forms object for each step in the test results tree.

For more information on running tests or components, refer to the *QuickTest Professional User's Guide*.

4

Testing .NET Windows Forms Applications

You can use the .NET Add-in to test .NET Windows Forms applications as well as custom .NET controls. You can also create checkpoints and output values on .NET Windows Forms objects to verify that the objects in your application function as expected.

| This chapter describes: | On page: |
|---|-----------------|
| About Testing .NET Windows Forms Objects | 44 |
| Recording Tests or Components on .NET Windows Forms Objects | 45 |
| Checking .NET Windows Forms Objects and Outputting Values | 48 |
| Using .NET Windows Forms Objects, Methods, and Properties to Enhance Your Test or Component | 50 |
| Running Tests or Components on .NET Windows Forms Objects | 51 |
| Using the .NET Windows Forms Spy | 52 |

About Testing .NET Windows Forms Objects

You can use the .NET Add-in to test .NET Windows Forms applications. You can also test most custom .NET controls inherited from the **System.Windows.Forms.Control** regardless of which language was used to create the application (for example, VisualBasic.NET, C#, and so forth).

You create and run steps on .NET Windows Forms applications similar to the way you do with standard Windows applications.

You can use the Keyword View and Expert View to activate .NET Windows Forms test object and run-time object methods, retrieve and set the values of properties, and check that objects exist.

You check the properties of a .NET Windows Forms object in the same way as you check the properties of corresponding Windows objects. When you create a checkpoint on a .NET Windows Forms object, QuickTest stores the selected property values of the object. If your application changes, you can modify the captured values to match the new expected values.

When you run a test or component, QuickTest verifies the behavior of your application by comparing values stored with the checkpoint to actual property values in your application. If the values do not match, the checkpoint fails. The results of the checkpoint can be viewed in the QuickTest Test Results window. For additional information on the Test Results window, refer to the *QuickTest Professional User's Guide*.

Note that this chapter provides examples using both the Keyword View and the Expert View. Some of the information provided and procedures described require working in the Expert View. For information on working in the Expert View, refer to the *QuickTest Professional User's Guide*.

Note: QuickTest Professional .NET Add-in Extensibility enables you to support testing on third-party and custom .NET controls that are not supported out-of-the-box by the QuickTest Professional .NET Add-in. For more information, refer to the *QuickTest Professional .NET Add-in Extensibility Developer's Guide*.

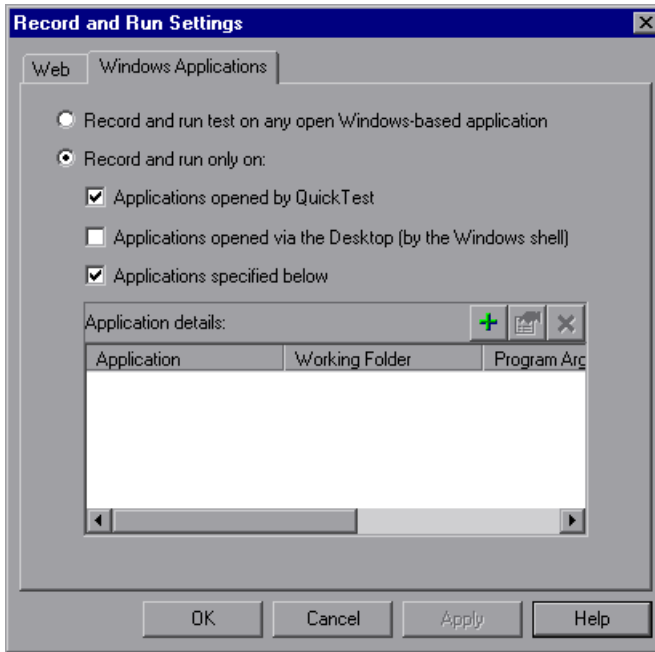
Recording Tests or Components on .NET Windows Forms Objects

You record steps on .NET Windows Forms objects similar to the way you record on corresponding standard Windows objects. For more information, refer to the *QuickTest Professional User's Guide*.

The Record and Run Settings dialog box opens automatically each time you begin recording a new test (unless you open the dialog box and set your preferences manually before you begin recording). You can also open it by choosing **Automation > Record and Run Settings**.

Note: The Record and Run Settings dialog box applies only to tests. Record settings for components are specified in the Applications tab or Applications dialog box of the relevant application area. However, specific record and run settings do not need to be defined for components. For more information on the Applications tab and Applications dialog box, refer to the *QuickTest Professional for Business Process Testing User's Guide*.

You can use the Windows Applications tab of the Record and Run Settings dialog box to instruct QuickTest to record only on applications opened by QuickTest or via the Windows Desktop, or .NET Windows Forms applications you specify, or you can instruct QuickTest to record on any open application.



Note: If you select **Record and run test on any open Windows-based application**, QuickTest also records any operations that you perform on standard Windows applications (such as Windows Explorer or an e-mail application) during the recording session.

For more information on the Record and Run Settings dialog box, refer to the *QuickTest Professional User's Guide*.

If you loaded other add-ins when you opened QuickTest, there may be additional tabs in the Record and Run Settings dialog box. If this is the case, confirm that the option to record and run on any open application (upper radio button of each tab) is selected.

While these settings do not directly affect your record or run sessions when working with .NET Windows Forms applications, they prevent QuickTest from opening unnecessary applications when you record or run tests or components on .NET applications.

When you record tests or components on a .NET Windows Forms application, QuickTest recognizes .NET Windows Forms objects. QuickTest inserts a step with the relevant .NET icon in the Keyword View and generates .NET-specific statements in the Expert View.

For example, if you record a click on a .NET Windows Forms button, the Keyword View may be displayed as follows:

| Item | Operation | Value | Documentation |
|-------------------|-----------|-------|--------------------------|
| ▼ Action1 | | | |
| ▼ SamplesExplorer | | | |
| ▼ Layouts | | | |
| Swap | Click | | Click the "Swap" button. |

QuickTest records these steps in the Expert View as:

```
SwfWindow("SamplesExplorer").SwfWindow("Layouts").SwfButton("Swap").
Click
```

For more information on .NET Windows Forms test objects, methods, and properties, refer to the .NET Windows Forms section of the *QuickTest Professional Object Model Reference*. For more information on creating tests and components, refer to the *QuickTest Professional User's Guide*.

Checking .NET Windows Forms Objects and Outputting Values

You use checkpoints to check the properties of .NET Windows Forms objects the same way you check the properties of Standard Windows objects. You can also output property values from the objects in your .NET Windows Forms application to use in your test or component.

The following checkpoints and output values are supported when testing .NET Windows Forms objects:

| Type | Checkpoint | Output Value |
|----------------|------------|--------------|
| Standard | Yes | Yes |
| Page | No | No |
| Accessibility | No | No |
| Text/Text Area | Yes | Yes |
| Table | Yes | Yes |
| Bitmap | Yes | No |
| XML | No | No |

You can check or output any test object property associated with an object using a standard checkpoint. For a list and description of the test object properties associated with each .NET Windows Forms object, refer to the .NET Windows Forms section of the *QuickTest Professional Object Model Reference*.

To check properties that are not included in the Checkpoint Properties dialog box, use the **Object** property. For more information, see “Accessing Internal Properties and Methods of Run-Time .NET Windows Forms Objects” on page 50.

For more information on checkpoints and output values, refer to the *QuickTest Professional User's Guide*.

Checking .NET Windows Forms Tables and Outputting Their Values

You check or output values from supported .NET Windows Forms grid controls using the Table Checkpoint Properties dialog box.

For tables with more than 100 rows, you can specify the rows you want to include in the checkpoint or output value in the Define Row Range dialog box. If you do not specify the rows to include, the table checkpoint or output value captures all data in the current level or view as follows:

| When working with: | The table checkpoint or output value captures: |
|--|---|
| ComponentOne C1FlexGrid and C1TrueDBGrid | The entire grid. |
| Microsoft Data Grid and DataGrid View | The currently displayed table (parent or child). |
| Infragistics UltraWinGrid | The band in which a cell, column, or row is selected. |
| DevExpress XtraGrid | The view that was most recently set. Tip: Insert a SetView method before your table checkpoint to ensure that the view you want is displayed when the table checkpoint runs. |

Apart from the difference in captured information as listed above, you define a table checkpoint or output value for .NET Windows Forms in the same way as you do for any other table. For more information, refer to the *QuickTest Professional User's Guide*.

Note: For a list of supported .NET Windows Forms grid control versions, refer to the *QuickTest Professional .NET Add-in Readme*.

Using .NET Windows Forms Objects, Methods, and Properties to Enhance Your Test or Component

A test or component consists of statements coded in Microsoft VBScript. These statements are composed of objects, methods, and/or properties that instruct QuickTest to perform operations or retrieve information. You add these statements using objects from your object repositories, and methods and properties that are available for each object type. In addition, when you record, these statements are generated automatically in response to input to the application. You can also program statements manually, or mix recorded and programmed statements in the same test or component. You create, view, and edit these statements in the Keyword View and/or Expert View.

Working with Test Object Methods

QuickTest provides a variety of test object methods that you can use with .NET Windows Forms objects. You can record some of these methods while recording on .NET Windows Forms objects. You can add additional functionality to your test or component by entering statements manually in the Keyword View or Expert View.

The .NET Add-in supports IntelliSense and statement completion in the Expert View, including the display of run-time (native) methods and properties when using the **Object** property in a statement. The .NET Add-in also supports generating statements that access run-time methods and properties in the Step Generator.

For more information, refer to the *QuickTest Professional User's Guide*.

Accessing Internal Properties and Methods of Run-Time .NET Windows Forms Objects

You can use the **Object** property to retrieve internal (native) properties and activate internal methods of any .NET Windows Forms object in your application.

For example, you can set the focus to a particular button and change its caption using statements similar to the following:

```
Set theButton = SwfWindow("frmWin").SwfButton("OK").Object
theButton.SetFocus
theButton.Caption = "Yes"
```

The **Object** property is also useful for verifying the value of properties that are not available using a standard checkpoint.

For additional information on the **Object** property and for information on .NET Windows Forms test objects and methods, refer to the .NET Windows Forms section of the *QuickTest Professional Object Model Reference*.

Running Tests or Components on .NET Windows Forms Objects

You run tests or components containing .NET Windows Forms objects in the same way as you run any other test or component. You can use the Windows Applications tab of the Record and Run Settings dialog box to instruct QuickTest to open your .NET Windows Forms applications each time you begin running your test, or you can instruct QuickTest to run only on open applications.

The test results tree displays the same .NET Windows Forms icons as those used in the Keyword View. If you choose to save still images or movies to the test results (**Tools > Options > Run** tab), the bottom right pane of the Test Results window displays the page that was captured during the run session and highlights the .NET Windows Forms object for each step in the test results tree.

For more information on running tests or components, refer to the *QuickTest Professional User's Guide*.

Using the .NET Windows Forms Spy

The .NET Windows Forms Spy enables you to view details about selected .NET Windows Forms controls and their run-time properties. You can use the .NET Windows Forms Spy to help you develop extensibility for .NET Windows Forms controls.

To spy on a .NET Windows Forms application, make sure that the application is running with Full Trust. If the application is not defined to run with Full Trust, you cannot spy on the .NET application's Windows Forms controls with the .NET Windows Forms Spy. For information on defining trust levels for .NET applications, refer to Microsoft documentation.

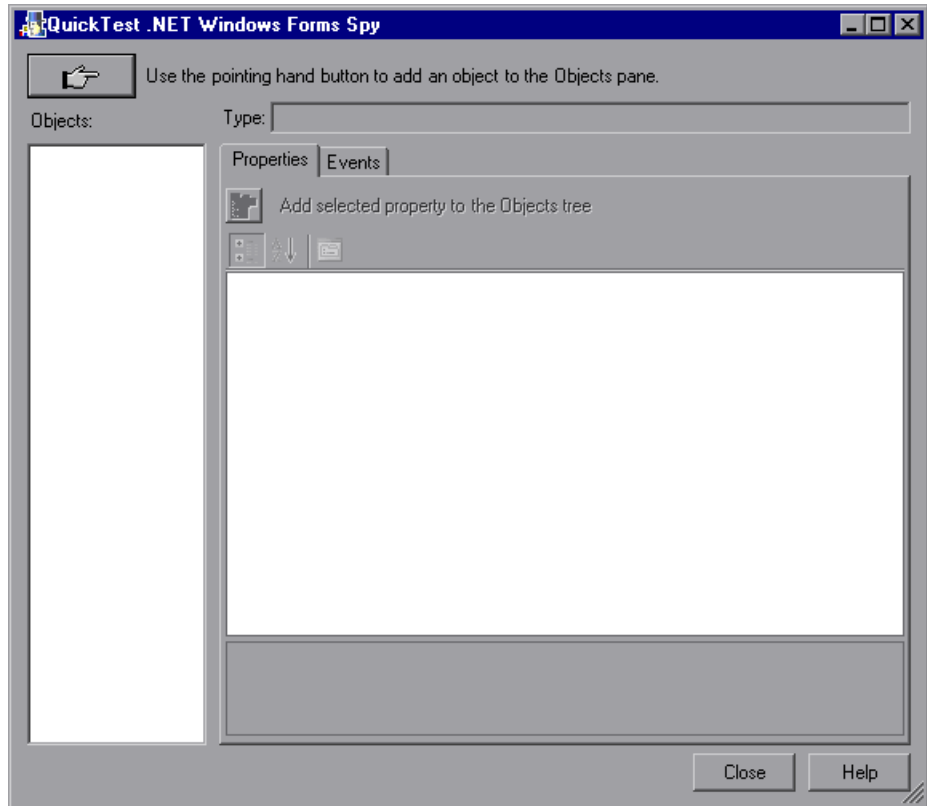
The .NET Windows Forms Spy is intended for advanced QuickTest users, especially those who are using .NET Add-in extensibility to create support for custom .NET Windows Forms controls. The .NET Windows Forms Spy can assist you in examining .NET Windows Forms controls within your application and seeing which events cause it to change (to facilitate recording and running) and how the changes manifest themselves in the control's state.

The .NET Windows Forms Spy enables you to select a specific control in your .NET application, view its run-time properties and values, change property values in the application in run-time, listen to events on a specific control, view the event arguments, and fire events back at the application.

Note: The .NET Windows Forms Spy runs in the context of your .NET application, not in the QuickTest context. The objects and run-time properties on which you are spying are the raw .NET controls in your application, and not the .NET test objects used in QuickTest. Since the .NET Windows Forms Spy runs in the context of your .NET application, you can close QuickTest while you use the .NET Windows Forms Spy. However, QuickTest must be open if you want to use the pointing hand mechanism to spy on additional objects. If you close the .NET application on which you are spying, the QuickTest .NET Windows Forms Spy window is closed automatically.

To open the .NET Windows Forms Spy and spy on an object:

- 1 Open the .NET Windows Forms application to the window containing the object on which you want to spy.
- 2 Choose **Tools > .NET Windows Forms Spy**. The QuickTest .NET Windows Forms Spy window opens.



- 3 In the QuickTest .NET Windows Forms Spy window, click the pointing hand. Both QuickTest and the .NET Windows Forms Spy are minimized so that you can point to and click on any object in the open application.

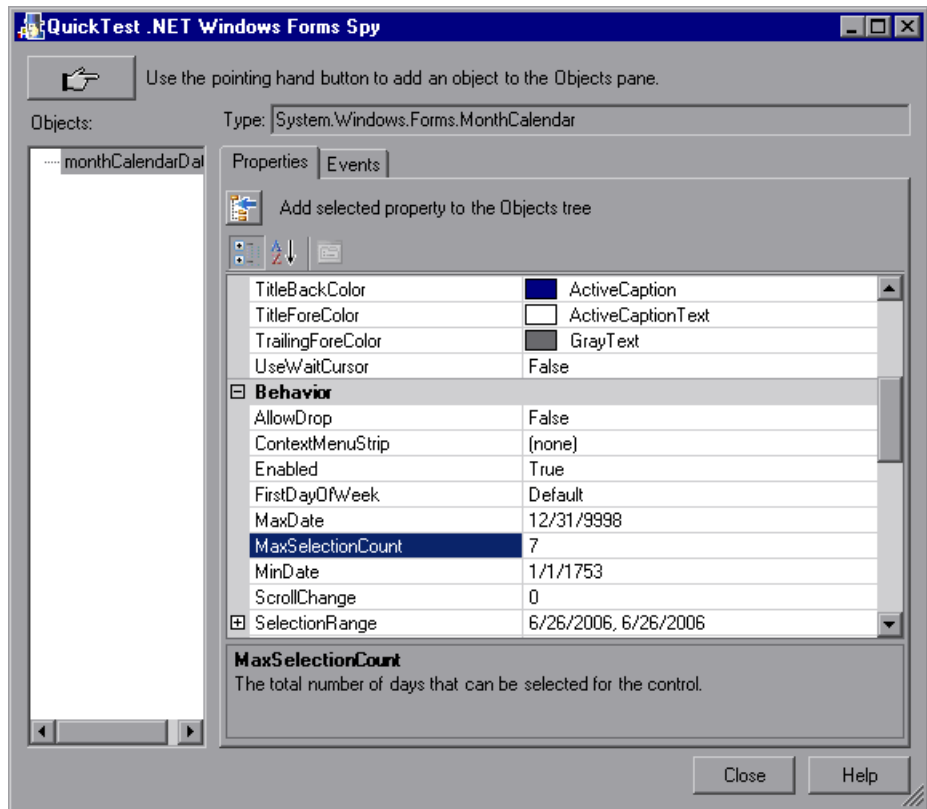
Tips: If the window on which you want to spy is partially hidden by another window, hold the pointing hand over the partially hidden window for a few seconds. The window comes into the foreground. You can now point and click on the object you want. You can configure the length of time required to bring a window into the foreground in the General tab of the Options dialog box. For more information, refer to the *QuickTest Professional User's Guide*.

You can also hold the left CTRL key to change the window focus. Additionally, if the window containing the object you want to select is minimized, you can display it by holding the left CTRL key, right-clicking the application in the Windows task bar, and choosing **Restore** from the context menu.


If the object on which you want to spy can be displayed only by performing an event (such as right-clicking or moving the pointer over an object to display a context menu), hold the left CTRL key. The pointing hand temporarily turns into a standard pointer and you can perform the event. When the object on which you want to spy is displayed, release the left CTRL key. The pointer becomes a pointing hand again.

- 4** Click the object whose properties you want to view. If the location you clicked in your application is associated with more than one object, the Object Selection dialog box opens. The objects associated with the location you clicked are displayed in hierarchical order.

- 5 Select the .NET Windows Forms object on which you want to spy and click **OK**. The QuickTest .NET Windows Forms Spy window opens showing the properties and values for the selected object.



The QuickTest .NET Windows Forms Spy window includes the following:

| Item | Description |
|--|---|
| <p>Pointing hand button</p>  | <p>Enables you to select a .NET Windows Forms object on which to spy. You can spy on as many objects within a single .NET application as you want. Each object that you select is added to the Objects pane.</p> <p>Note: If you select an object from a different .NET application, an additional QuickTest .NET Windows Forms Spy window opens, showing the information for the selected object.</p> |
| <p>Type</p> | <p>Displays the full type name of the selected object.</p> |
| <p>Objects pane</p> | <p>Displays a hierarchical tree of the objects you selected to spy. For more information, see “Working with the Objects Pane” on page 57.</p> |
| <p>Properties tab</p> | <p>Enables you to view and modify run-time property values in your .NET application. For more information, see “Working with the Properties Tab” on page 58.</p> |
| <p>Events tab</p> | <p>Enables you to listen to events in your .NET application and fire them at the application. For more information, see “Working with the Events Tab” on page 61.</p> |

- 6 You can repeat steps 3 to 5 to spy on additional objects and add them to the Objects pane in the QuickTest .NET Windows Forms Spy window.

Note: QuickTest must be open if you want to use the pointing hand mechanism to spy on additional objects.

You can now view and modify the object’s run-time property values. In addition, you can view, listen to, and fire events on the object.

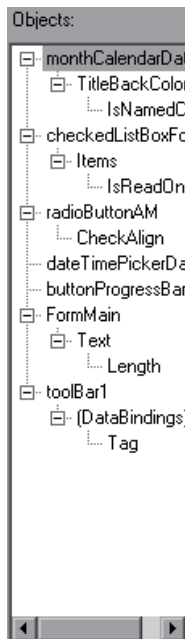
Working with the Objects Pane

The Objects pane contains a list of the objects in your .NET application on which you have spied. Each time you spy on another object in the same .NET application, it is added to the Objects pane. You can spy on as many objects from the same .NET application as you want, using the pointing hand button in the QuickTest .NET Windows Forms Spy window.

The Objects pane also contains any embedded objects that you added from the Properties tab. Each time you add an embedded object to the Objects pane, it is added below its parent object, in a hierarchical format. For more information on viewing properties for embedded objects, see “Working with the Properties Tab” on page 58.

You can select an object in the Objects pane and view or modify its properties and property values, and listen to and fire its events.

You can remove objects from the Objects pane if you no longer need them. You cannot delete the last remaining parent object from the Objects pane. When you remove an object, its descendants (if any) are also removed.



To remove objects from the Objects pane:

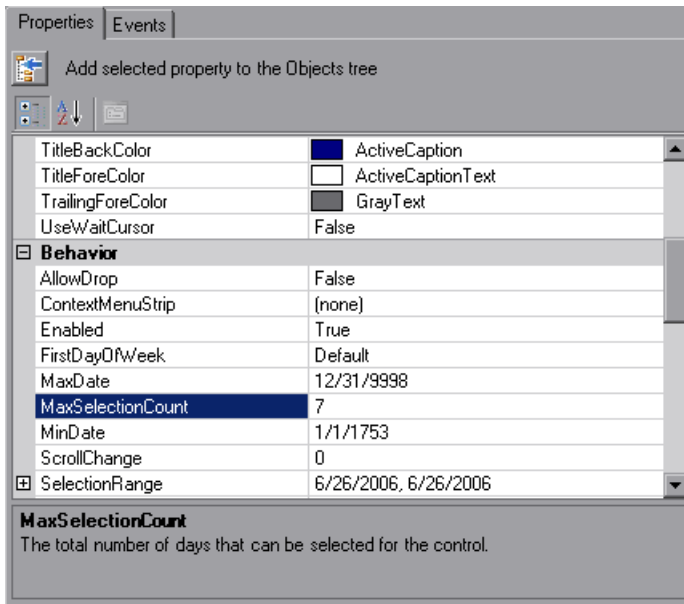
- 1 Select the object that you want to remove.
- 2 Perform one of the following:
 - Right-click the object and choose **Remove Object**.
 - Press DELETE.

Working with the Properties Tab

The Properties tab enables you to view run-time properties and values for objects in your .NET application. You can select a property to display a description of the property below the property grid. You can choose to display the properties alphabetically or by category.

You can change property values in the .NET Windows Forms Spy and apply those changes to your .NET application in run-time.

You can also add embedded objects from the Properties tab to the Objects pane to view their properties.



To view run-time property values for a .NET Windows Forms object:

In the Objects pane, select the object whose run-time properties you want to view. The properties for the selected object are displayed in the Properties tab, with the property names on the left, and the property values on the right. A description of the selected property is displayed below the properties grid.

To modify run-time property values for a .NET Windows Forms object:

- 1** In the Properties tab, click the property value you want to modify. Properties shown in gray are defined as read-only in the .NET application and cannot be modified.
- 2** Edit the property value as required. The property value displays different types of edit fields, depending on the needs of a particular property. These edit fields include edit boxes, drop-down lists, and links to custom editor dialog boxes.

After you modify a property value, the new value is applied to the run-time instance of the .NET application. For example, you can change the text of an edit box label, change the background color of a dialog box from gray to red, and so forth.

Note: Any changes you make to the run-time property values of the .NET application remain in effect only for the current instance of the .NET application. The next time you run the .NET application, the properties will return to their original run-time values.

To view properties of embedded objects:

- 1** In the Properties tab, select the property whose embedded object properties you want to view.
- 2** Click the **Add Selected Property** button. The property is added to the Objects pane, and its run-time properties and property values (if any) are shown in the Properties tab. Each time you add an embedded object to the Objects pane, it is added below its parent object, in a hierarchical format.



Note: The **Add Selected Property** button is disabled if the property's value is null, or the property is an object with no properties of its own.

To sort the properties grid:

Click one of the following buttons to sort the properties grid in the Properties tab:



- ▶ **Categorized.** Lists all properties and property values for the selected object, by category. Categories are listed alphabetically. You can collapse a category to reduce the number of visible properties. When you expand or collapse a category, a plus (+) or minus (-) is displayed to the left of the category name.



- ▶ **Alphabetical.** Alphabetically sorts all run-time properties for the selected object.

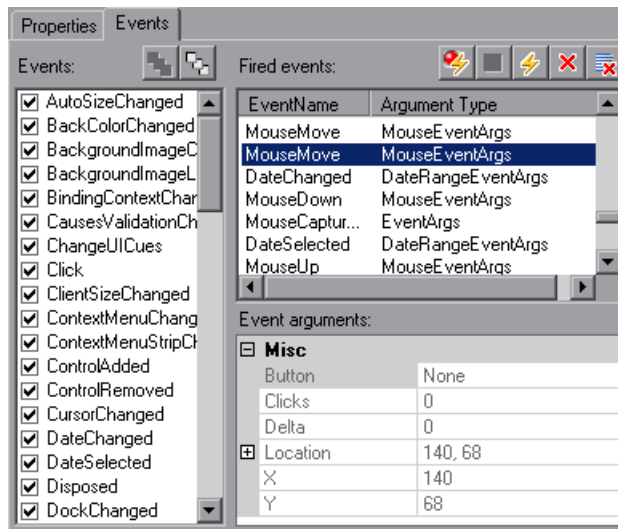


Note: The **Property Pages** button is not currently supported.

Working with the Events Tab

The Events tab enables you to listen to selected events on a specific control in your .NET application. You can then view the event arguments, and fire selected events back at the application.

This is especially useful if you are using .NET Add-in extensibility to create support for custom .NET Windows Forms controls. You can see which events cause your .NET application to change, so you can implement extensibility for recording operations on specific controls, and also check which events need to be fired to make your .NET application behave the way you want.



To listen to specified events for a .NET Windows Forms object:

- 1** In the Objects pane, select the object to whose events you want to listen.
- 2** In the Events list, select the check boxes for the event types to which you want to listen.

Note: The events that you select affect only the events that are listened to and logged by QuickTest. If you select or clear a check box for an event type after listening to events for an object, the events in the Fired Events list are not changed.



Tip: You can click the **Select All Events** or **Clear All Events** buttons to select or clear all the event check boxes. You can also right-click the Events list and choose **Select All** or **Clear All**.



- 3** Click the **Listen to Selected Events** button. QuickTest starts listening to the specified events on the selected object.
- 4** In your .NET application, perform the operations on the object to whose events you want to listen. The specified events are logged as they occur and are shown in the Fired Events list.



- 5** When you want to stop listening to events, click the **Stop Listening to Events** button. QuickTest stops listening to and logging the specified events.

To view event arguments for a .NET Windows Forms object:

- 1** In the Objects pane, select the object whose event arguments you want to view.
- 2** Select the event in the Fired Events list whose arguments you want to view. The selected events arguments and argument values are shown directly below the event, in the Event Arguments list.

To fire selected events on a .NET Windows Forms object:

- 1** In the Objects pane, select the object whose events you want to fire.
- 2** In the Fired Events list, select one or more events that you want to fire on your .NET application. You can select multiple events using standard Windows selection techniques (CTRL and SHIFT keys).

Tip: The selected events are fired in the order in which they appear in the Fired Events list. If the events do not appear in the Fired Events list in the order in which you want to fire them, listen to more events on the object until the events you want are added to the Fired Events list in the required order.

- 3 If the events you selected have editable arguments, you can change their argument values in the Event Arguments list if needed before firing the events. When the events are fired, they will be fired with the modified argument values.



- 4 Click the **Fire Selected Events** button. The selected events are fired in the order in which they appear in the Fired Events list. You can view the effect that firing these events has on the relevant object in your .NET application.

To remove specific events from the Fired Events list:

- 1 In the Objects pane, select the object whose events you want to remove from the Fired Events list.
- 2 Select the events in the Fired Events list that you want to remove. You can select multiple events using standard Windows selection techniques (CTRL and SHIFT keys).



- 3 Click the **Clear Selected Events** button. The selected events are removed from the Fired Events list.

To clear all events from the Fired Events list:

- 1 In the Objects pane, select the object whose events you want to remove from the Fired Events list.



- 2 Click the **Clear Event List** button. All the logged events are removed from the Fired Events list.

5

Testing Windows Presentation Foundation Applications

You can use the .NET Add-in to test WPF (Windows Presentation Foundation) applications. You can also create checkpoints and output values on WPF objects to verify that the objects in your application function as expected.

| This chapter describes: | On page: |
|--|-----------------|
| About Testing WPF Objects | 66 |
| About WPF User Interface Automation | 67 |
| Recording Tests or Components on WPF Objects | 68 |
| Checking WPF Objects and Outputting Values | 71 |
| Using WPF Objects, Methods, and Properties to Enhance Your Test or Component | 72 |
| Running Tests or Components on WPF Objects | 74 |

About Testing WPF Objects

You can use the .NET Add-in to test WPF applications. You can also test most custom WPF controls inherited directly or indirectly from the **System.Windows.Controls.Control** class regardless of which language was used to create the application (for example, VisualBasic.NET, C#, and so forth), as well as third-party WPF controls that are inherited from the **System.Windows.Controls.Control** class and implement automation interfaces.

WPF uses UI (User Interface) Automation to define UI objects. UI Automation provides standardization of controls and properties for the functionality of objects. The .NET Add-in supports UI Automation through the AutomationElement and AutomationPattern properties.

You create and run steps on WPF applications similar to the way you do with standard Windows applications.

You can use the Keyword View and Expert View to activate WPF test object, Automation object and run-time object methods, retrieve and set the values of properties, and check that objects exist.

You check the properties of a WPF object in the same way as you check the properties of corresponding Windows objects. When you create a checkpoint on a WPF object, QuickTest stores the selected property values of the object. If your application changes, you can modify the captured values to match the new expected values.

When you run a test or component, QuickTest verifies the behavior of your application by comparing values stored with the checkpoint to actual property values in your application. If the values do not match, the checkpoint fails. The results of the checkpoint can be viewed in the QuickTest Test Results window. For additional information on the Test Results window, refer to the *QuickTest Professional User's Guide*.

Note that this chapter provides examples using both the Keyword View and the Expert View. Some of the information provided and procedures described require working in the Expert View. For information on working in the Expert View, refer to the *QuickTest Professional User's Guide*.

About WPF User Interface Automation

UI Automation provides a single, consistent, reference object for UI elements in multiple frameworks (For example, Win32, WPF, and Trident). With UI Automation, the functionality of objects in the UI is defined by a set of standard controls and properties that are common to all objects of that type.

To learn more about UI Automation, see the UI Automation Fundamentals page of the Microsoft Developer Network library at <http://msdn2.microsoft.com/en-us/library/ms753107.aspx>.

Automation Elements

UI Automation exposes every element in the UI as an **Automation Element**. Automation Elements expose common properties of the UI elements they represent.

For example, a button control has the **Automation Element** property **NameProperty**, which references the name or text associated with a button control. That same property is called **caption** or **alt** in Win32 and HTML, respectively. With UI Automation, all button controls have a **NameProperty**, which is mapped to the corresponding property in each framework.

The **Automation Element** also exposes **control patterns** that provide properties and expose methods specific to their control types.

Control Patterns

Control patterns represent discrete pieces of functionality that a control in the UI can perform. The total set of control patterns for a control type define the functionality of that control type.

Control patterns expose **methods** that provide the ability to programmatically manipulate the control.

Control patterns expose **properties** that provide information about the control's functionality and current state.

The set of supported control patterns for a particular control can be dynamically defined. Therefore, a particular control type may not always support the same set of control patterns. For example, a multiline edit box supports scrolling (**scrollpattern** pattern) only if its text exceeds the viewable area.

Some controls types, such as Image controls do not support any control patterns.

QuickTest Professional enables you to access the methods and properties of automation elements and control patterns using special properties in the QuickTest object model for WPF.

For information on how to work with UI Automation in your test or component, see “Accessing Internal Properties and Methods of WPF Objects” on page 72.

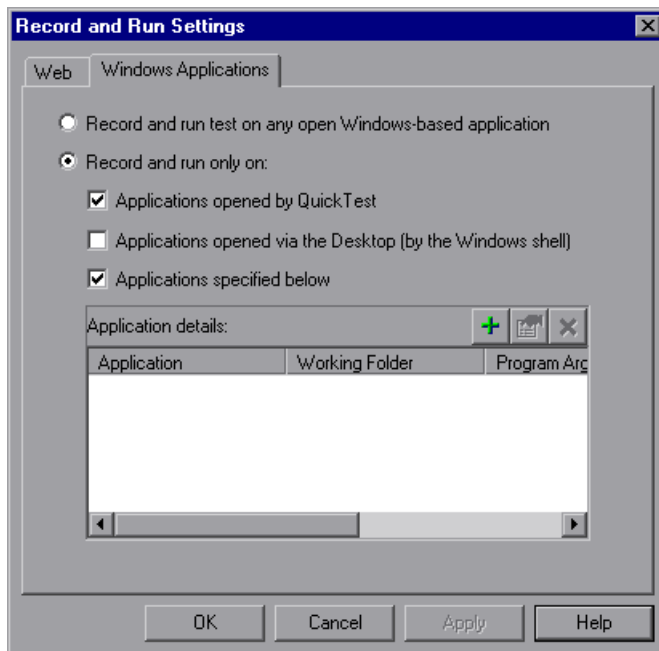
Recording Tests or Components on WPF Objects

You record steps on WPF objects similar to the way you record on corresponding standard Windows objects. For more information, refer to the *QuickTest Professional User's Guide*.

The Record and Run Settings dialog box opens automatically each time you begin recording a new test (unless you open the dialog box and set your preferences manually before you begin recording). You can also open it by choosing **Automation > Record and Run Settings**.

Note: The Record and Run Settings dialog box applies only to tests. Record settings for components are specified in the Applications tab or Applications dialog box of the relevant application area. For more information on the Applications tab and Applications dialog box, refer to the *QuickTest Professional for Business Process Testing User's Guide*.

You can use the Windows Applications tab of the Record and Run Settings dialog box to instruct QuickTest to record only on applications opened by QuickTest, via the Windows Desktop, by WPF applications you specify, or you can instruct QuickTest to record on any open application.



Note: If you select **Record and run test on any open Windows-based application**, QuickTest also records any operations that you perform on standard Windows applications (such as Windows Explorer or an e-mail application) during the recording session.

For more information on the Record and Run Settings dialog box, refer to the *QuickTest Professional User's Guide*.

If you loaded other add-ins when you opened QuickTest, there may be additional tabs in the Record and Run Settings dialog box. If this is the case, confirm that the option to record and run on any open application (upper radio button of each tab) is selected.

While these settings do not directly affect your record or run sessions when working with WPF applications, they prevent QuickTest from opening unnecessary applications when you record or run tests or components on WPF applications.

When you record tests or components on a WPF application, QuickTest recognizes WPF objects. QuickTest inserts a step with the relevant WPF icon in the Keyword View and generates WPF-specific statements in the Expert View.

For example, if you record a click on a WPF button, the Keyword View may be displayed as follows:

| Item | Operation | Value | Comment | Documentation |
|----------------|-----------|-------|---------|-----------------------------|
| ▼ Action1 | | | | |
| ▼ SimpleStyles | | | | |
| Default | Click | | | Click the "Default" button. |

QuickTest records this step in the Expert View as:

```
WpfWindow("SimpleStyles").WpfButton("Default").Click
```

For more information on WPF test objects, methods, and properties, refer to the **Windows Presentation Foundation** section of the *QuickTest Professional Object Model Reference*. For more information on creating tests and components, refer to the *QuickTest Professional User's Guide*.

Checking WPF Objects and Outputting Values

You use checkpoints to check the properties of WPF objects the same way you check the properties of standard Windows objects. You can also output property values from the objects in your WPF application to use in your test or component.

The following checkpoints and output values are supported when testing WPF objects:

| Type | Checkpoint | Output Value |
|----------------|------------|--------------|
| Standard | Yes | Yes |
| Page | No | No |
| Accessibility | No | No |
| Text/Text Area | Yes | Yes |
| Table | No | No |
| Bitmap | Yes | No |
| XML | No | No |

You can check or output any test object property associated with an object using a standard checkpoint. For a list and description of the test object properties associated with each WPF object, refer to the WPF section of the *QuickTest Professional Object Model Reference*.

To check properties that are not included in the Checkpoint Properties dialog box you can use the **Object**, **AutomationElement** or **AutomationPattern** property. For more information, see “Accessing Internal Properties and Methods of WPF Objects” on page 72.

For more information on checkpoints and output values, refer to the *QuickTest Professional User’s Guide*.

Using WPF Objects, Methods, and Properties to Enhance Your Test or Component

A test or component consists of statements coded in Microsoft VBScript. These statements are composed of objects, methods, and/or properties that instruct QuickTest to perform operations or retrieve information. You add these statements using objects from your object repositories, and methods and properties that are available for each object type. In addition, when you record, these statements are generated automatically in response to input to the application. You can also program statements manually, or mix recorded and programmed statements in the same test or component. You create, view, and edit these statements in the Keyword View and/or Expert View.

Accessing Internal Properties and Methods of WPF Objects

When accessing the internal properties and methods of WPF objects, it is important to know which property to use to access the object that contains the information you want to set or retrieve.

- ▶ **AutomationElement property.** Returns the object that gives access to the set of standard properties that expose information about the **Automation Element**.
- ▶ **AutomationPattern property.** Returns the object that gives access to the specific instance of a **Control Pattern**.
- ▶ **Object property.** Returns the object that gives access to properties specific to the actual run-time UI object, as defined by the developer.

Many of the properties and methods accessible through the **AutomationElement** and **AutomationPattern** properties contain the same information as the properties and methods accessible through the **Object** property. However, information available through UI Automation that is accessed through the **Object** property lacks the standardization provided by UI Automation.

Custom properties designed by the developer are accessible only through the **Object** property.

Working with Test Object Methods

QuickTest provides a variety of test object methods that you can use with WPF objects. You can record some of these methods while recording on WPF objects. You can add additional functionality to your test or component by entering statements manually in the Keyword View or Expert View. For information on the available WPF test objects, methods, and properties, see the **Windows Presentation Foundation** section of the *Object Model Reference*.

Note: Because WPF Automation Elements of the same control type may support a different set of control patterns, the test object methods or properties that QuickTest supports for a specific test object may be different than the standard set of methods and properties listed in the **Windows Presentation Foundation** section of the *Object Model Reference*. For example, although the WpfButton test object generally does not support the **Set** method, QuickTest may record a **Set** step when clicking a toggle button if the control pattern that was activated during the record session corresponds to the **Set** method.

The .NET Add-in supports IntelliSense and statement completion in the Expert View, including the display of available run-time (native) methods and properties when using the **Object** and **AutomationElement** properties in a statement. The .NET Add-in also supports generating statements that access run-time methods and properties in the Step Generator.

To use IntelliSense with the **Object** and **AutomationElement** properties, ensure that the application you are referencing is open and displays the object to which you are referring.

Running Tests or Components on WPF Objects

You run tests or components containing WPF steps in the same way as you run any other test or component. You can use the Windows Applications tab of the Record and Run Settings dialog box to instruct QuickTest to open your WPF applications each time you begin running your test, or you can instruct QuickTest to run only on open applications.

The test results tree displays the same WPF icons as those used in the Keyword View. If you choose to save still images or movies to the test results (**Tools > Options > Run** tab), the right side of the Test Results Window will display the captured images.

For more information on running tests or components, refer to the *QuickTest Professional User's Guide*.

Index

A

- accessing internal properties and methods of run-time .NET Web Forms objects 41
- accessing internal properties and methods of run-time .NET Windows Forms objects 50
- Add-in Manager dialog box 26
- add-ins
 - loading with add-in support 26
 - loading without add-in support 28
 - repairing 29
 - uninstalling external 30
- administrator privileges for installing 16
- Automation Elements 67
- AutomationElement property 72
- AutomationPattern property 72

C

- checkpoints
 - supported for Web Forms objects 40
 - supported for Windows Forms objects 48
 - supported for Windows Presentation Foundation objects 71
- components
 - running on .NET Web Forms objects 42
 - running on .NET Windows Forms objects 45
 - running on Windows Presentation Foundation objects 68
- control patterns 67, 73
- conventions. *See* typographical conventions
- custom .NET controls 35, 44
- Customer Registration screen 20
- customer support site 20

D

- disk space requirements 15
- documentation updates 11

E

- events
 - viewing with .NET Windows Forms Spy 61
- Expert View 41, 50, 73
- external add-ins, uninstalling 30

H

- hard disk space requirements 15

I

- installation requirements 15
- InstallShield Wizard Complete screen 24
- internal methods, accessing 41, 50

K

- Knowledge Base 10

L

- License Agreement screen 18
- loading QuickTest
 - with add-in support 26
 - without add-in support 28

M

- Mercury Best Practices 10
- Mercury Customer Support Web site 10
- Mercury Home Page 10

Index

methods 41, 50, 72
test object 41, 50, 73

N

native properties, accessing 41, 50
.NET Web Forms objects, testing 35
.NET Windows Forms Spy 52
.NET Windows Forms objects, testing 43

O

Object property 41, 50
online documentation 8
online resources 10
opening .NET applications during record 37, 45
opening Windows Presentation Foundation applications during record 68
output values
supported for Web Forms objects 40
supported for Windows Forms objects 48
supported for Windows Presentation Foundation objects 71

P

prerequisites, installation 15
properties
viewing with .NET Windows Forms Spy 58

Q

QuickTest
loading with add-in support 26
loading without add-in support 28

R

Readme 8
Record and Run Settings dialog box 37, 42, 45, 51, 68, 74
recording tests, on .NET objects 37, 45
recording tests, on Windows Presentation Foundation objects 68

repairing add-ins 29
requirements, installation 15
running tests
on .NET Web Forms applications 42
on .NET Windows Forms applications 51
on Windows Presentation Foundation applications 74

S

Select Updates screen 21
Send Feedback 10
setup
Customer Registration screen 20
InstallShield Wizard Complete screen 24
License Agreement screen 18
running 16
Spy
.NET Windows Forms 52
statements 41, 50, 72
Step Generator 41, 50, 73
system requirements 15
System.Windows.Forms.Control, supported controls 35, 44

T

test objects 41, 50, 72
Test Results window 42, 51, 74
tests
running on .NET Web Forms objects 42
running on .NET Windows Forms objects 45
running on Windows Presentation Foundation objects 68
typographical conventions 12

U

UI Automation
control patterns 67
uninstalling external add-ins 30
updates, documentation 11

V

VBScript 41, 50, 72

W

Windows Forms

 spying on 52

Windows Presentation Foundation

 AutomationElement property 72

 AutomationPattern property 72

 Object property 72

Windows Presentation Foundation objects,

 testing 65

