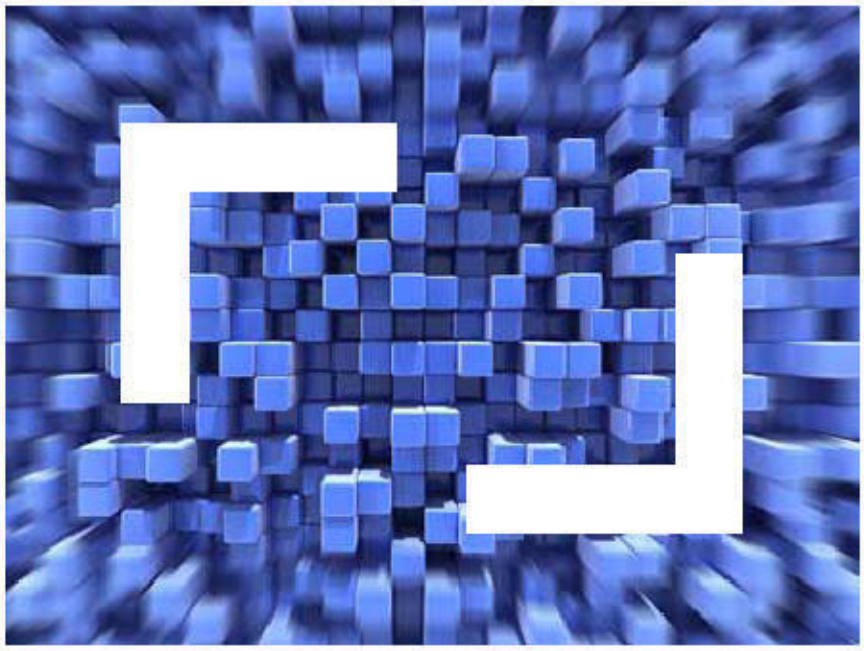


TestPartner®

Getting Started with Visual Tests

Borland®
(A MICRO FOCUS COMPANY)

MICRO FOCUS®
Leading the Evolution™

Borland Software Corporation
4 Hutton Centre Dr., Suite 900
Santa Ana, CA 92707

Contents

Welcome to the TestPartner Visual Test Tutorial

Welcome to the TestPartner Visual Test tutorial, a self-paced guide that demonstrates how to use TestPartner's visual, storyboard-based interface to create powerful and flexible functional tests. In this tutorial, you will learn the basic steps required to create a visual test, play back the visual test, and then analyze the results of the playback. Additionally, you will learn how to use a number of features that allow you to quickly update and enhance a recorded visual test.

This tutorial uses the sample Web application, *http://demo.borland.com/InsuranceWebExtJS/*, to create a real world scenario in which you practice using TestPartner to create repeatable tests.

The lessons in this tutorial are designed to be completed in sequence as each lesson is based on the output of previous lessons.

Introducing the GUI

This section introduces the GUI including the Main Screen, Start Screen, and Visual Navigator. This section is optional. If you are already familiar with the basic elements of the development environment, proceed to the next section.

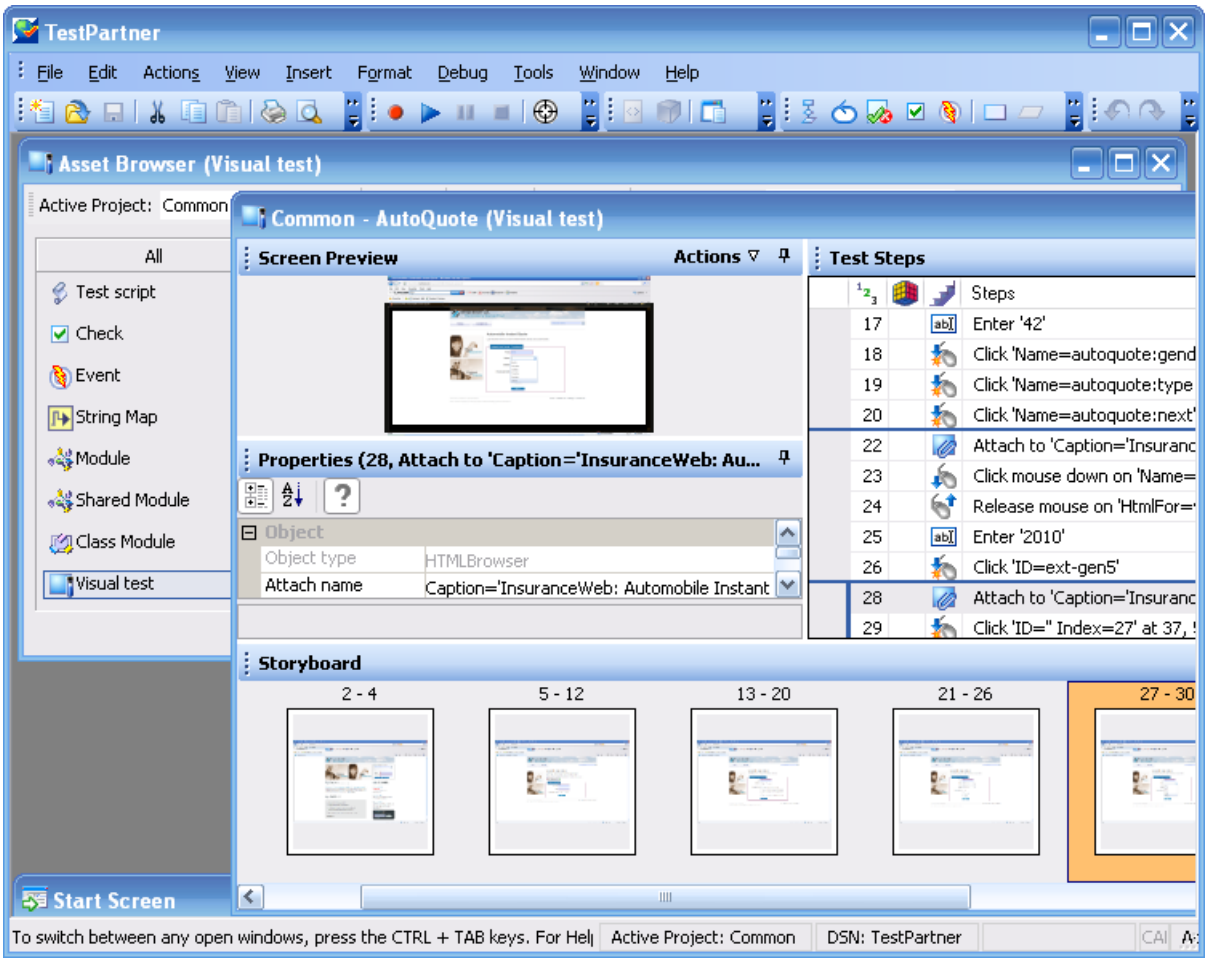
Main Screen

TestPartner's Main Screen is the starting point for all test activities. All other TestPartner windows, including the Start Screen, Visual Navigator, and Asset Browser, display as child windows in the TestPartner Main Screen.

The TestPartner Main Screen contains the following:

- Menu bar
- Toolbars
- Main asset viewing area
- Status bar

The following graphic shows these elements in the TestPartner Main Screen. The main asset viewing area shows a visual test in the Visual Navigator, the Asset Browser in the background, and the Start Screen, which is minimized.

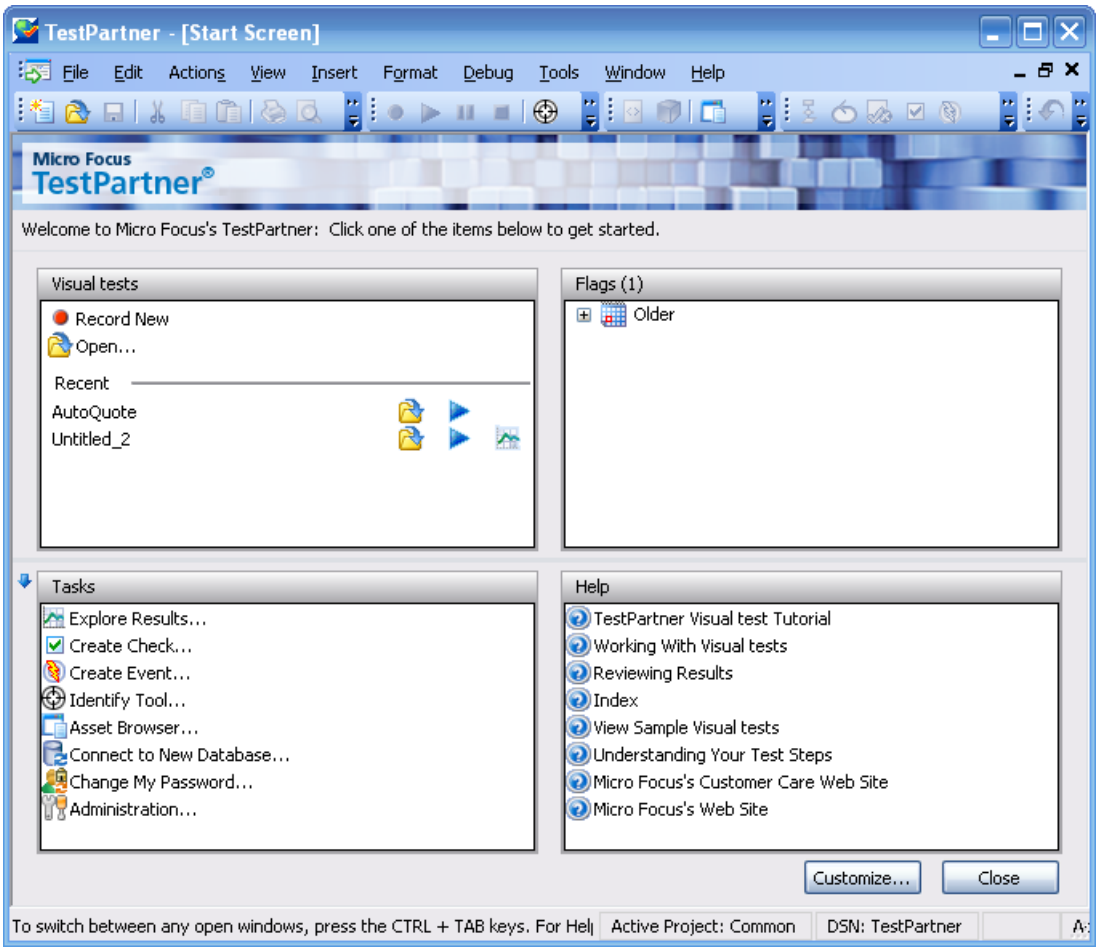


Start Screen

The Start Screen is the launching point for creating and managing visual tests, as well as links to all commonly performed testing activities in TestPartner. The Start Screen contains four panes:

- Visual tests
- Tasks
- Flags
- Help

The following graphic shows the Start Screen with the four panes displayed. The Visual tests pane shows recently accessed visual tests and the Flags pane shows flags that have been assigned to the tester.



- Use the **Visual tests** pane to access and record new visual tests. You can also open, play back, or view the results of the most recently accessed visual tests.
- Use the **Tasks** pane to access frequently used features.
- Use the **Flags** pane to collaborate test suite and test project information with other testers and users of the test projects in the database.
- The **Help** pane provides quick links to learning assistance to help get you productive quickly and lets you access product information right when you need it.

Click **Customize** to change the option to display the **Start Screen** when TestPartner starts.

Click **Close** at any time to hide the **Start Screen**.

Click **Collapse Bottom Panel** to hide and **Expand Bottom Panel** to show the **Tasks** and **Help** panes in the **Start Screen**.

Tip: You can always access it from the main screen by clicking the **Start Screen** in the watermark. You can also press **`Ctrl+Alt+S`**, or choose **View ➤ Start Screen** at any time to display it.

Note: The user interface works with standard Windows small and large font sizes. Using a custom font size may result in the inaccurate display of text in the user interface.

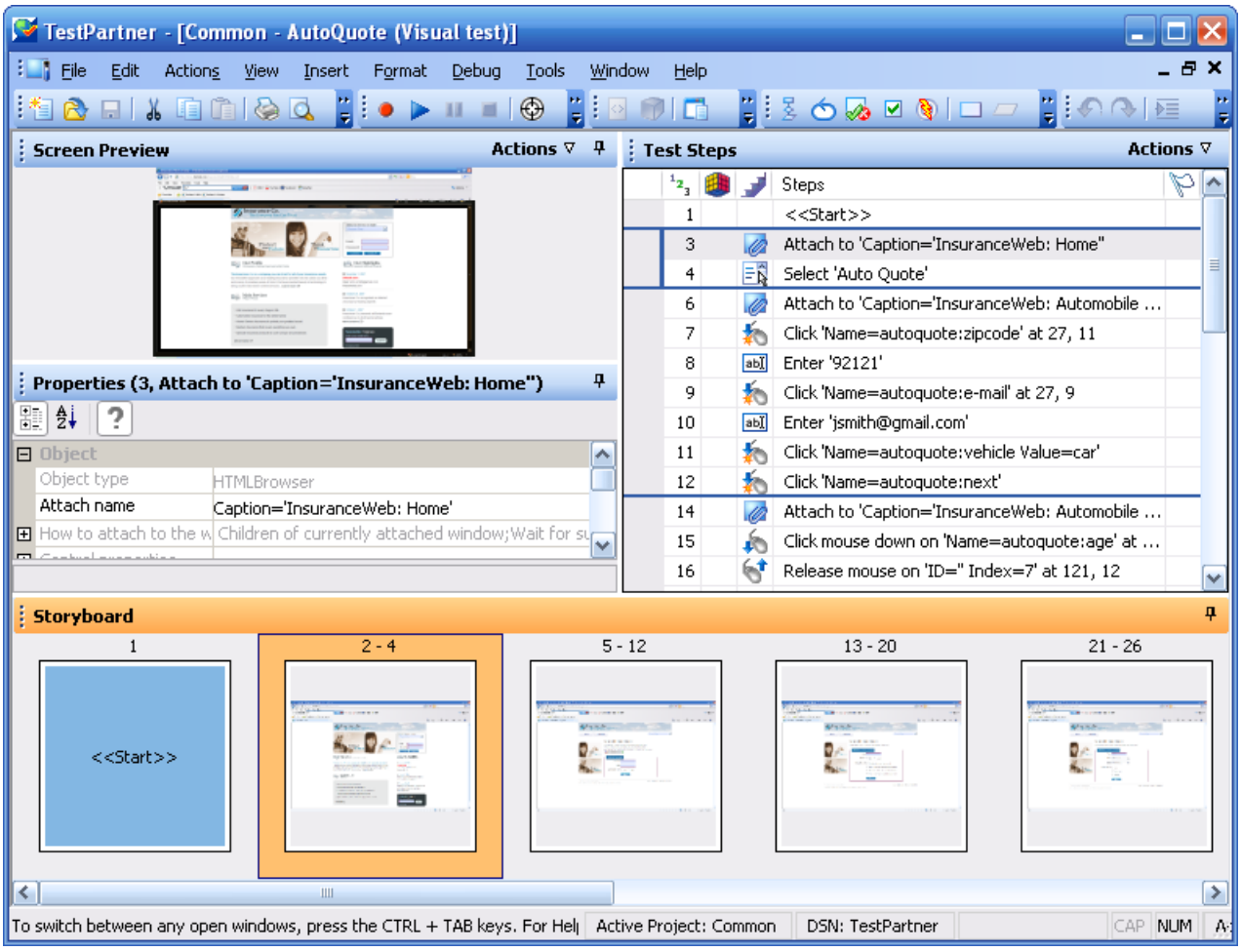
Next, learn about the Visual Navigator.

Visual Navigator

The Visual Navigator graphically represents a visual test. Use the Visual Navigator to interact with and edit test elements through a point-and-click interface. The Visual Navigator consists of four main areas:

- Test Steps pane
- Screen Preview
- Properties pane
- Storyboard

The following graphic shows a visual test in the Visual Navigator, which displays in the TestPartner main window.



The four panes include:

Test Steps	Lists each step of a visual test in clear non-technical language.
Screen Preview	Displays a snapshot of the application under test as it appears when a step executes during playback of a visual test.

Properties	Displays the properties of a step in a visual test.
Storyboard	Displays the flow of a visual test through the use of thumbnail images, which represent the logical groups of steps in a visual test.

The Screen Preview, Storyboard, and Properties panes are synchronized with the Test Steps pane and display information specific to a selected step in the Test Steps pane. In this way, you can easily view all aspects of a step by selecting a step in the Test Steps pane, and then viewing information about the step in the other panes.

In addition to viewing a visual test, the Visual Navigator also allows you to enhance or update an existing visual test by using the Screen Preview and Properties panes. For example, in the Properties pane, after recording a visual test, you can change the literal value of a recorded property by replacing it with a variable.

Additionally, to quickly update a visual test when changes occur in the application under test, you can update previously captured screens using the Update Screen feature of the Screen Preview.

The Visual Navigator also displays the playback result of a visual test using the same panes as those used for a visual test. For a result, the panes have additional functionality and appear in the Result window, which contains toolbar options and several tabs that display different views of result content. Examples of additional functionality specific to a result include the ability to see the pass or fail status of checks and verifications in the Test Steps pane. Additionally, in the Screen Preview, you can see a comparison of the differences between the screens captured during recording and screens captured during playback, and then update the existing visual test without accessing the test application.

After you have reviewed the basic elements of the TestPartner development environment, you are ready to begin recording a test.

Recording a Visual Test: Introduction

As you perform actions to create an insurance quote request in the sample Web application, TestPartner records the actions as steps. Steps form the basis of a visual test. When you have completed recording actions needed for a test, you can see the recorded test in the Visual Navigator. Steps display in the Test Steps pane of the Visual Navigator.

Starting the Sample Web Application

For this tutorial, use the Borland sample Web application. This Web application is provided for demonstration purposes.

Use the Borland sample Web application with Internet Explorer. To ensure a user experience consistent with the lessons in the tutorial, we do not recommend running the sample Web application with the Mozilla Firefox browser.

To access the sample application remotely, click *http://demo.borland.com/InsuranceWebExtJS*.
The sample application Web page opens.

Recording a Visual Test for the Sample Web Application

During recording, TestPartner records all interactions in the test application (except interaction with TestPartner itself) until recording is stopped. After you have finished recording, you can modify the visual test you have generated to add and remove steps.

1. From the TestPartner Start Screen, click **Record New** in the Visual tests pane.
   TestPartner minimizes and a pop-up message appears indicating that recording has started.
2. In the Insurance Company Web site, perform the following steps:

   During recording the TestPartner icon on the task bar flashes. You can see the current object that you are working with and the last action that was recorded in the **Recording** dialog box.

   a) From the **Select a Service or login** list box, select **Auto Quote**.
      The **Automobile Instant Quote** page opens.
   b) Type a zip code and email address in the appropriate text boxes, click an automobile type, and then click **Next**.

      To follow this tutorial step-by-step, type **`92121`** as the zip code, **`jsmith@gmail.com`** as the email address and specify **`Car`** as the autombile type.
   c) Specify an age, click a gender and driving record type, and then click **Next**.

      For example, type **`42`** as the age, specify the gender as **`Male`** and **`Good`** as the driving record type.
   d) Specify a year, make, and model, click the financial info type, and then click **Next**.

      For example, type **`2010`** as the year, specify **`Lexus`** and **`RX400`** as the make and model, and **`Lease`** as the financial info type.

      A summary of the information you specified appears.
   e) Click **Purchase**.

      The **Purchase A Quote** page opens.
   f) Click **Home** near the top of the page to return to the home page where recording started.
3. Press **`Alt+F10`** to complete recording.
   The **Recording Complete** dialog box opens.

Saving and Naming the Visual Test

Once you have recorded your test, the **Recording Complete** dialog box opens and then you can:

- Play back the recorded visual test.
- Review the recorded actions in the Visual Navigator.
- Save the visual test, and then review the recorded test in the Visual Navigator.

Since you have just recorded the visual test for the first time, save and name the test before playing it back or reviewing the recorded actions.

1. From the **Recording Complete** dialog box, click **Save**.

   Tip: When you create an asset without naming it, TestPartner assigns the temporary name *"Untitled_"* followed by a sequential number.

   Because the test has not been named yet, the **Save As** dialog box opens.
2. In the **Name** text box, change the name to **`AutoQuote`**.
3. In the **Description** text box, type **`Visual test tutorial`**.
4. Click **OK**.
   The visual test displays in the Visual Navigator.

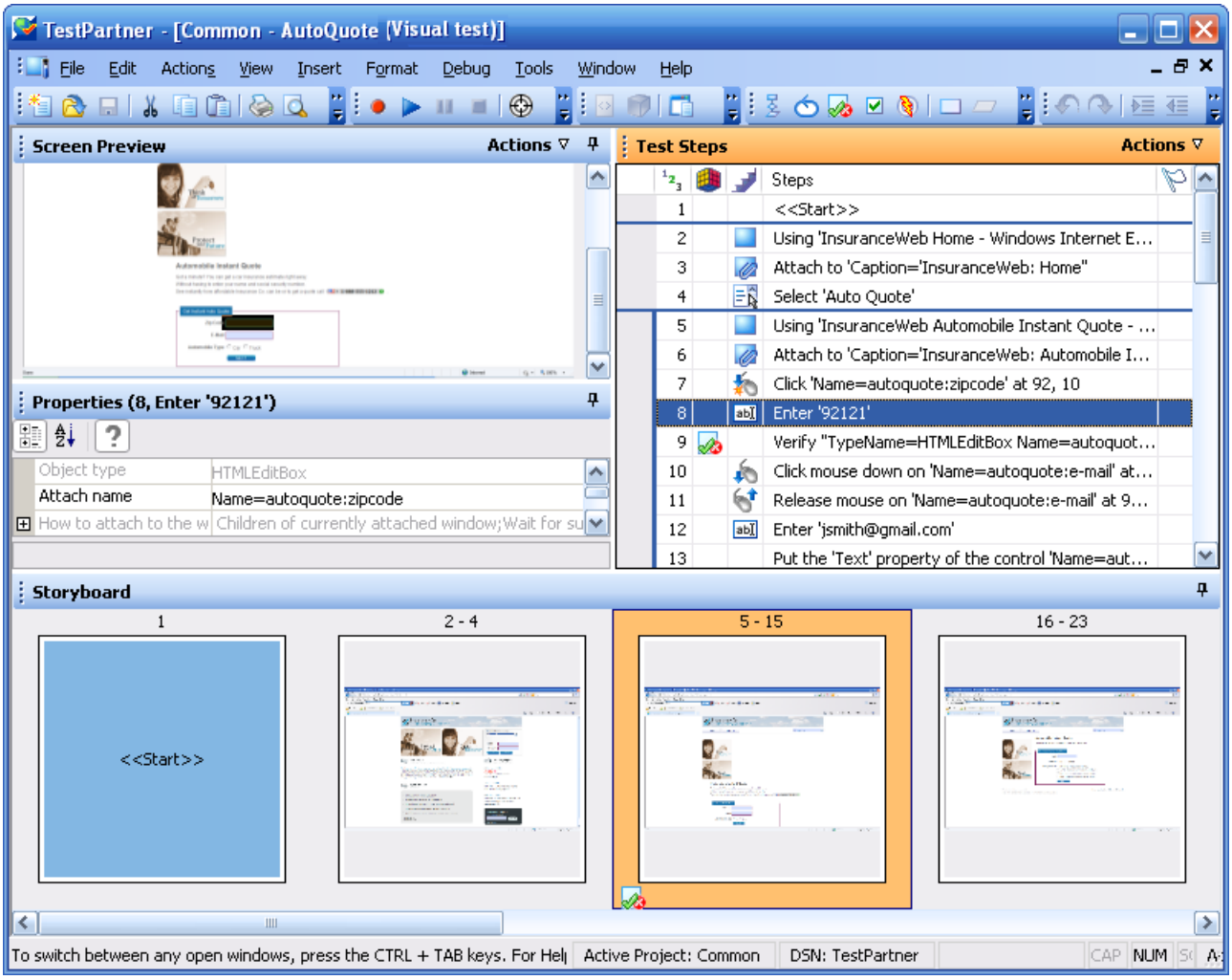
Reviewing the Recorded Test Steps

Once you have recorded your visual test and saved it, the visual test displays in the Visual Navigator. The four panes of the Visual Navigator provide a comprehensive view of the visual test.

Steps in the Test Steps pane represent the screens accessed during recording and the actions performed while completing the quote in the Web application.

Note: By default, TestPartner does not display steps which set focus to screens in the test application. This reduces the total number of steps in the visual test and increases the readability of a visual test. For this tutorial, turn on the option to view all steps. From the Test Steps pane title bar, choose **Actions ➤ View ➤ Steps and Screens**.

Your recorded steps should be similar to the steps in the following graphic.



The columns in the Test Steps pane include:

Column	Description
	Represents the sequential order in which steps are recorded and played back.

Column	Description
	If you change the view from the default view (Steps only) to the Screens only or the Steps and Screens view, the test steps are not renumbered. As a result, there might be gaps in the numerical sequence.
	Displays icons representing the type of logic for the step, if the step contains logic. See *Test Step Logic Types* in the online help for more information on the different logic icons and their meaning.
	• Displays icons representing the type of action being executed in the step. See *Test Step Icons* in the online help for more information about icons that represent types of steps. • The icons representing step types you'll see in the recorded visual test include: • Screen step: Sets focus to the specified window or controls in a target application to interact with it or any of its child controls. • Attach step: Attaches to the specified control to allow interaction with the control. • Automation step: Simulates a mouse click on the specified control. • Automation step: Simulates a mouse double-click on the specified control. • Automation step: Enters a specified string into a text-based control. • Automation step: Set the text of an EditBox control using encrypted text. This type of step is normally recorded for an action that enters a password into an EditBox control.
Steps	Plain-text description of the action being taken for the step. See *Test Step Types* in the online help for descriptions of different types of step descriptions.
	Displays either the Flag icon or the Assigned Flag icon .
	Displays a user-defined step description. This column does not display in the default view, but can be shown by clicking **Actions ➤ View ➤ Step Description**.

Tip: Different panes in the Visual Navigator are synchronized with the **Test Steps** pane. In the preceding graphic, the recorded step that enters the zip code is selected in the **Test Steps** pane. As a result:

- The **Screen Preview** shows the state of the application before the zip code is entered.
- The **Properties** pane shows the properties for the selected step.
- The thumbnail representing the group of steps related to typing the zip code is highlighted in the Storyboard.

Scroll through the steps in the visual test and select various steps to view the updated information in the other panes.

Playing Back the Recorded Visual Test

Once you have recorded and saved your visual test, you can play it back to verify that the visual test works.

1. Perform one of the following steps:
   - Choose **Actions ➤ Playback**.
   - Click **Playback** on the toolbar.

The **Playback** dialog box opens. This dialog box lets you determine how the result is saved.

2. In the **Result description** text box, type **`Initial test results for the recorded test`**.
3. Click **OK**.
   TestPartner minimizes and the visual test plays back. During playback, the actions you performed while recording the visual test are played back on the screen against the sample application. When playback completes successfully, the **Playback Complete** dialog box opens.

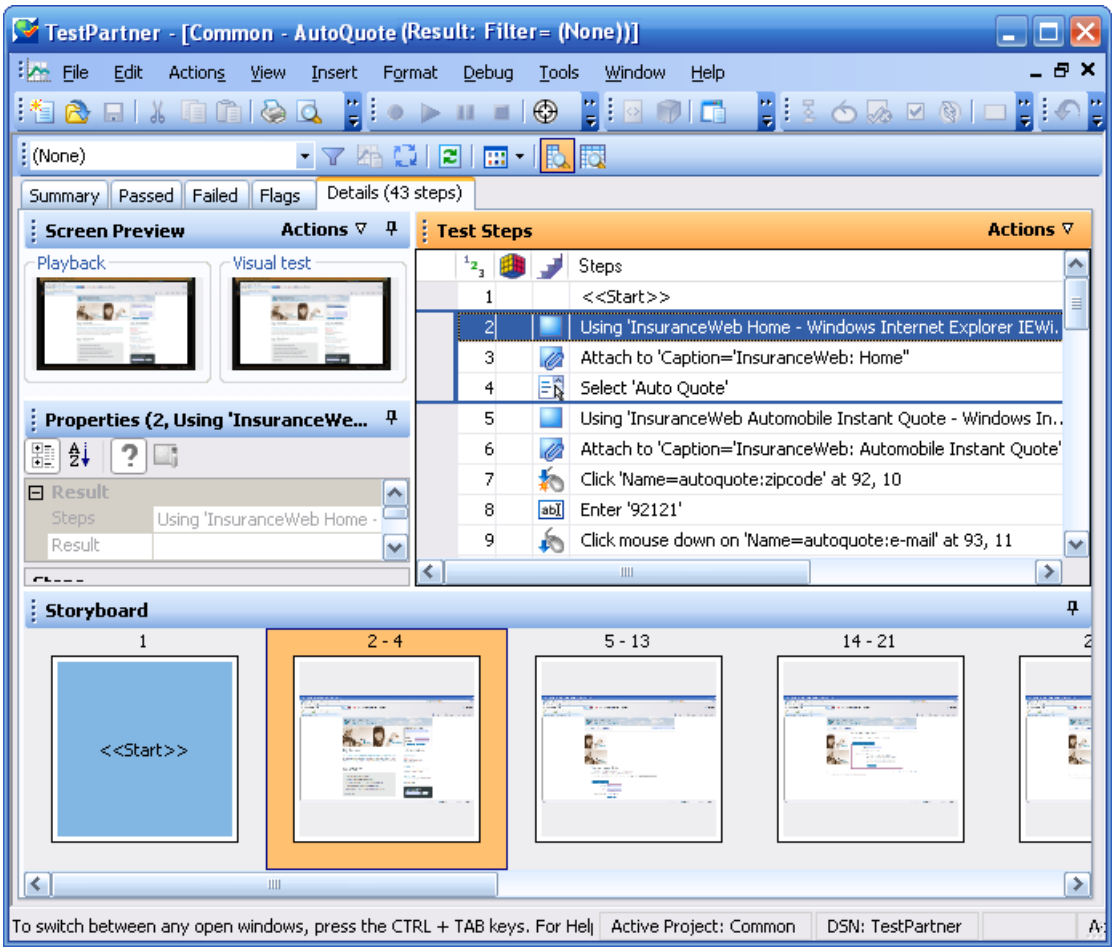
Analyzing Results: Introduction

After playing back a visual test, TestPartner generates a test result. A test result contains information about the playback of the visual test. Information such as the name of the visual test, the run number, the date and time each step executed, the pass or fail status of each step, and other important information.

Result Window Overview

After playing back a visual test, TestPartner displays the playback results in the Result window. The Result window contains three main areas that help you analyze all aspects of the test results. The three areas are:

- Result window toolbar: Provides options for customizing the display and type of content found in a result.
- Result window tabs: Organizes and displays result information into pre-defined categories. Use the tabs in the **Result** window to quickly find specific result information.
- Visual Navigator: Graphically represents test results using four panes: Screen Preview, Test Steps, Properties, and Storyboard.



Result Window Toolbar

Customize the display and type of content found in a result. Use the toolbar options to quickly and easily perform the following:

- Apply filters
- Set a pass criteria to define the success and failure of checks and verifications
- Show the details of each test run
- Refresh the result
- View only steps and/or screens
- Select a basic view or advanced view of the result information

Result Window Tabs

Use the tabs in the **Result** window to quickly find specific result information. The tabs function as filters that organize and display result information as follows:

- Summary tab: Displays a high-level overview report containing information such as whether the playback was successful, the latest run number, the number of checks or verifications that passed or failed, the start time and end time of the test, and other basic information about the result of the test run.
- Passed tab: Displays all passed checks or verifications .
- Failed tab: Displays all failed checks or verifications.

- Flags tab: Displays any flagged steps. Remind yourself or other testers about important issues by inserting a flag in any step of a result or by using verification logic to create and insert a flag during playback.
- Details tab: Displays each step of a visual test. Information such as the name of the test step, pass/fail status of checks or verifications, a description, and flag status.

Note: The Passed, Failed, Flags, and Details tabs all graphically represent the result of test runs using the four panes of the Visual Navigator.

Visual Navigator Panes

For visual tests, the Passed tab, Failed tab, Flags tab, and Details tab all contain the four panes of the Visual Navigator:

- Test Steps
- Screen Preview
- Properties
- Storyboard

The Properties pane contains a Show/Hide Step Properties of Visual Test Before Playback toolbar button, which allows you to show or hide the visual test properties of a selected step.

For scripts, the **Result** window displays only the Test Step and Properties panes. The Test Steps pane provides more information about the playback status and the result of each test step in the Command Detail column.

To customize the display of result data, you can modify the default result view from the General options. You can modify playback results options from the Playback Results options. Additionally, you can create filters to view desired data from the **Result** window.

Using the Result Window Tabs

To quickly view test result information, the **Result** window contains five tabs which function as filters that organize and display specific types of result information.

1. Click **Go to Result** on the **Playback Complete** dialog box.

   The **Result** window opens to the **Summary** tab by default. The **Summary** tab provides an overview of the test run including information such as whether the playback was successful or not, the latest run number, the number of verifications that passed or failed, the start time and end time of the test, and other basic information about the result of the test run.
2. Click the **Details** tab.

   The **Details** tab displays the result of every step using the four panes of the Visual Navigator:

Test Steps	Lists information about the playback result of each step in the visual test.
Screen Preview	Displays the Web application screens captured during playback.
Properties	Displays the properties of a step.
Storyboard	Provides a graphical outline and overview of a result.

Note: The **Passed**, **Failed**, and **Flags** tabs also display result information using the Visual Navigator. The only difference is that these tabs display specific types of steps, whereas, the **Details** tab displays every step.

3. Select any step.

   TestPartner updates the other panes with information specific to the selected step. In the **Screen Preview**, the screen captured during playback is compared against the screen captured when the visual test was first recorded. In the **Properties** pane, the properties of the selected step are listed. And in the **Storyboard**, the group of steps in which the step occurs is highlighted.

   Tip: To view the entire name of the step, you might have to expand the **Steps** column or move your pointer over the step to display a ToolTip containing the entire name.

Using the Result Window Toolbar

The **Result** window toolbar provides several options for customizing the display and type of content found in a result.

1. Click **Advanced View** on the Results toolbar.
   TestPartner displays additional columns in the **Test Steps** pane and additional properties in the **Properties** pane.
2. In the **Test Steps** pane, scroll to the right to view the additional columns.

   These columns provide specific information about each step such as, the time in milliseconds it took for the step to run, the user name of the person who ran the test, and other specific details.
3. In the **Properties** pane, the Result property category lists the corresponding properties of each column in the **Test Steps** pane. Scroll down to view the entire list.
4. Click the **View** drop-down arrow and select **Screens Only**.
   TestPartner filters out all the non-screen steps, so you can quickly see only the screen steps.
5. Click the **View** drop-down arrow and select **Steps and Screens**.
   TestPartner displays all the screens and steps of the result.

   The **Result** window toolbar contains additional options for customizing the display and type of content found in a result.

Using the Properties Pane

The **Properties** pane displays properties of a step that describe the basic characteristics of the step including the name of the step, the execution status, the details about the playback performance, and other information.

1. Click the step `Enter '42'`.

   This step performs the action of typing the value '42' into the **Age** text box.

   The **Properties** pane updates and displays the properties of the selected step.
2. Click the **Categorized** icon if it is not already selected.

   The properties are grouped into three main categories:

Result	Contains properties that correspond to the columns in the **Test Steps** pane. Properties such as the name of the step, the date and time the step was executed, and the time it took to run the step.

Playback details	Contains properties and their values captured from controls in the Web application during playback of the test.
Extended properties	Contains additional playback details such as the attach name TestPartner uses to identify the control or the value of the text entered by an `Enter` action step. Extended properties are helpful to view the contents of variables or expressions when they are used in action steps. For example, an `Enter` step that uses the variable `textVar` as its value displays the contents of `textVar` in the Extended properties category.

3. Expand the **Extended properties** node.
   The **Text** property and **Attach name** property appear with their respective values listed. The **Text** property value is 42, which is the value entered in the **Age** text box for this step.
4. Click the **Show/Hide Step Properties of Visual Test Before Playback** icon.
   The Visual test details node appears and lists all of the original properties of the step as they exist in the visual test.

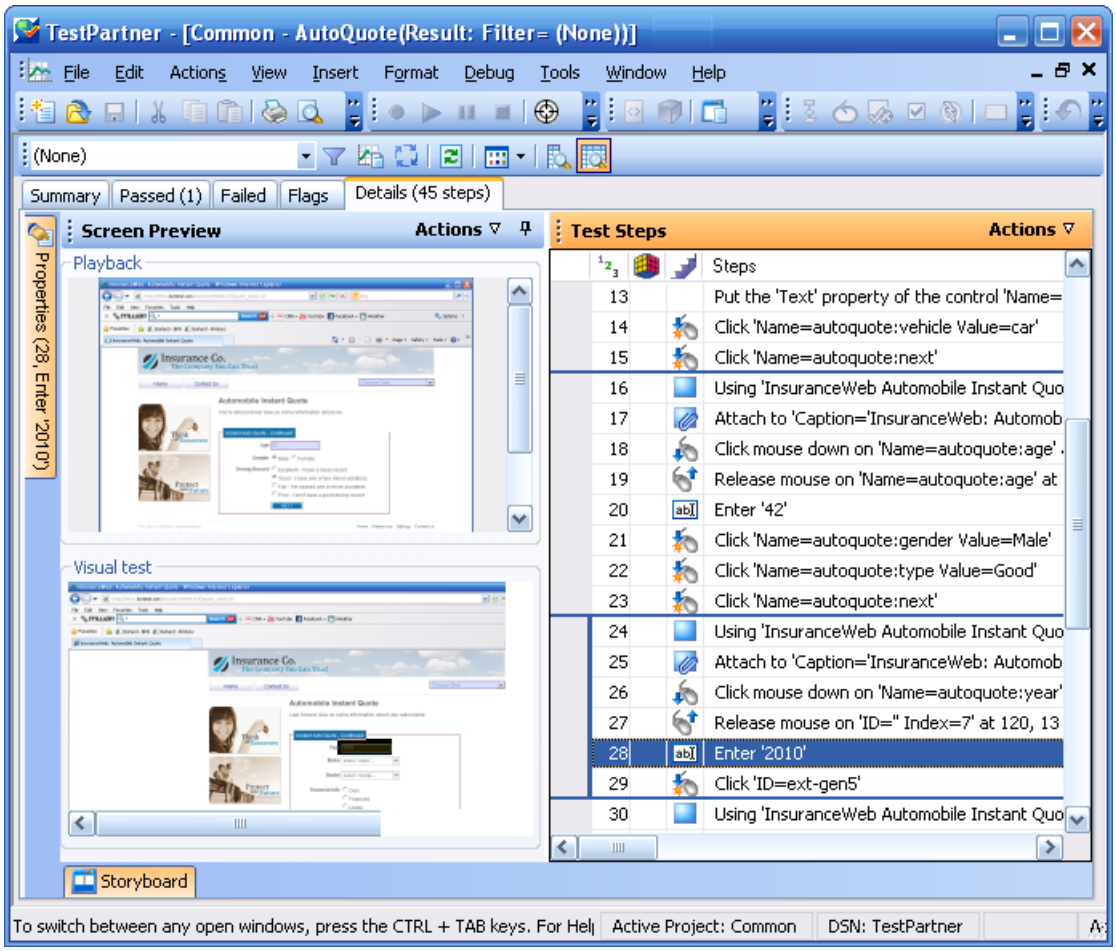
Using the Screen Preview

The **Screen Preview** displays a captured image of the test application for each step in the **Test Steps** pane that interacts with a control. The captured image can show the full desktop, the application window, or only the active window. Captured images represent the state of the application before its associated step is executed.

1. In the **Test Steps** pane, select the step that contains the year of the car.

   ```
   Enter '2010'
   ```

   In the **Screen Preview**, the screen captured during playback appears next to the screen captured during the recording of the visual test. The control for this step, the edit box, is highlighted by a black box.



2. In the **Screen Preview**, click **Actions ➤ Show Differences ➤ Off**.
   The Visual test screen closes and the Playback screen opens.
3. Click **Actions ➤ Zoom ➤ 75%**, then use the scroll bars to position the page so that it displays the year clearly.
4. Switch to the visual test. In the **Test Steps** pane, click **Actions ➤ Visual test Window**.

Enhancing the Visual Test: Introduction

Enhancing a visual test includes making updates to the existing visual test to ensure that it works with newer versions of the test application. For example, to handle and verify varying conditions in the test application you can insert test logic. Additionally, to increase the readability of a visual test or to remind yourself or others about important aspects of the test, you can insert a flag, message box, or step description.

These are just some of the ways in which you can use TestPartner to enhance an existing visual test to create more powerful, robust, and flexible tests.

Updating From the Screen Preview

When TestPartner records a visual test, it captures screens from the test application in addition to the associated controls for each screen. In the **Screen Preview**, TestPartner displays each captured screen and highlights the control that is identified by a given step in the visual test. From the **Screen Preview**, you can update a step by identifying a different control in the captured screen without accessing the test application.

In this lesson, you will select a different button from the captured screen in the **Screen Preview** using TestPartner's **Insert Control From Screen Preview** feature.

1. In the **Test Steps** pane of the AutoQuote visual test, select the step before the <<End>> step.

   The step text is similar to the following:

   ```
   Click 'Caption=Home'
   ```

   The **Screen Preview** displays the **Purchase A Quote** page and highlights the **Home** button.
2. In the **Properties** pane, click the **Attach name** column.
   The property value selection buttons display in the value area of the property.
3. In the **Screen Preview**, click **Actions ➤ Zoom ➤ 75%**, then use the scroll bars to position the screen so that the **Contact Us** button is clearly visible.
4. In the **Properties** pane, click **Identify from the screen preview**.
   The pointer moves to the **Screen Preview**.
5. Move the pointer over the **Contact Us** button and then click the **Contact Us** button.

   When you click the button, the **Attach name** property in the **Properties** pane changes to display the new attach name for the button, and the step text in the **Test Steps** pane changes to the following:

   ```
   Click 'Caption='Contact Us"
   ```
6. Click **Save**.

The next time you play back the visual test, TestPartner clicks the **Contact Us** button instead of the **Home** button.

Inserting a Verification

A verification is test logic that evaluates a user-defined condition, and then sends a pass/fail message and, optionally, a flag to the playback result of a visual test.

In this lesson, you will insert a verification to ensure that the quote uses the correct zip code.

1. Ensure that you are viewing both steps and screens by clicking **Actions ➤ View ➤ Steps and Screens** in the **Test Steps** pane.
2. Select the `Enter '92121'` step.
3. In the **Screen Preview**, click **Actions ➤ Zoom ➤ 75%**, then use the scroll bars to position the screen so that the **Zip Code** is clearly visible.
4. Perform one of the following steps:

   - Click **Create Verification Type Logic** on the toolbar.
   - Choose **Insert ➤ Test Logic ➤ Verification**.

   The **Welcome** page of the **Test Logic Designer** wizard opens.
5. Click **Next**.
   The **Select a Logic Type** page opens.

6. Click **The property of a control**, and then click **Next**.
   The **Define a property-based condition** page opens.
7. Click **Identify from the screen preview**.
   The pointer moves to the **Screen Preview**.
8. Click the **Zip Code** text box.
   The **Define a property-based condition** page displays the name of the selected control, in this case, the zip code, and the properties of the control.

   The **Select the condition** is set to **Is Equal to** while the **Expected value** is empty.
9. In the **Select a property** section, select **Text**.
10. In the **Expected value** text box, type **92121**.
11. Click **Next**.
    The **Build the Verification** page opens. From this page, you can define the pass or fail message to send to the playback result. At the top of the page, the condition that TestPartner verifies should be as follows:

    ```
    "TypeName=HTMLEditBox Name=autoquote:zipcode"."Text" Is Equal to "92121"
    ```

    This condition defines the logic for the verification. The condition compares the zip code to the zip code entered.
12. Replace the default pass text description to **`The zip code is correct`**, and the default failed text description to **`The zip code is NOT correct`**.
13. Click **Next**.
    The **Summary** page opens.

    A verification step is inserted after the selected step. The step text for the verification step is similar to the following:

    ```
    Verify "TypeName=HTMLEditBox Name=autoquote:zipcode"."Text" Is Equal to "92121"
    ```
14. After reviewing the verification, click **Finish**.

You have successfully enhanced the recorded visual test by inserting test logic that verifies the value of a property in the sample application.

Creating a Local Variable to Store Application Data

Variables enhance visual tests by providing the ability to store data values for use in other parts of the test or in other visual tests or test scripts. Data can also be output to other types of files.

In this lesson, you will store variable text in the control that displays the email address so it can be used in a later lesson of the tutorial. To do this, you must first create a local variable to store the text.

1. In the **Test Steps** pane, click **Actions ➤ Insert ➤ Variable ➤ Add Local**.
   The **Add Local Variable** dialog box opens.
2. In the **Variable name** text box, type **`strEmailAddress`**.
3. From the **Type** list, select **Text**.

   Leave the **Initial value** text box empty for this lesson, since a value will be stored to the variable in a subsequent lesson.

   The `text` type stores the variable value as a text, or string data type.
4. Click **OK**.
   The new variable is saved for the visual test. Once the variable is created, you can see it and edit its definition from the `<<Start>>` step.
5. To view the *strEmailAddress* variable, select the `<<Start>>` step in the **Test Steps** pane.

Tip: The <<Start>> step is always the first step in any visual test.

Properties for the step display in the **Properties** pane. With the <<Start>> step selected, *strEmailAddress* displays in the **Variables** category defined as a **Text** variable.

The value area for the variable types indicate the number of variables of the type that are currently defined. In this lesson, one Text variable has been created, so the value area for the item shows that one item is associated with this test.

Now that you have created the local variable to store the quote email address, store the email address text from the application to the variable.

Storing Application Data to the Local Variable

The sample application displays a unique email address on the **Get Instant Auto Quote page** page. The text on this page containing the email address is the property value of a control on the page.

In this lesson, you will store this text to the local variable *strEmailAddress* created in the previous exercise.

1. In the **Test Steps** pane, select the step that shows the email address value.

   The step text should look similar to the following:

   ```
   Enter 'jsmith@gmail.com'
   ```

   When the step is selected, the captured screen for the **Get Instant Auto Quote page** page displays in the **Screen Preview**.

2. In the **Test Steps** pane, click **Actions ➤ Insert ➤ Property from Control**.

   This inserts a step into the visual test just after the selected step. The step text should be similar to the following:

   ```
   Put the '' property of the control " into "
   ```

   This step will be used to store the email address text from the **Get Instant Auto Quote page** page to the local variable by editing the step properties. You will edit the step properties to specify the control to be used, the property of the control, and the variable to store the property value. The variable is *strEmailAddress*, which you created in the previous exercise.

3. In the **Screen Preview**, click **Actions ➤ Zoom ➤ 50%**, and then use the scroll bars to position the screen so that the email address text is clearly visible.
4. In the **Properties** pane, click the **Attach name** column.
   The property value selection buttons display in the value area of the property.
5. Click **Identify from the screen preview**.
   The pointer moves to the **Screen Preview**.
6. Move the pointer over the **E-Mail** text on the page.

   Ensure the highlighted box appears around the text on the page, then click the highlighted area to identify the control.

   TestPartner updates the **Object** properties information in the **Properties** pane with the following value:

   ```
   Put the '' property of the control 'Name=autoquote:e-mail' into "
   ```

7. In the **Properties** pane, click the **Property name** column, and then select **Text** from the list for its value.
8. Click **(Select a local variable...)** and then select **strEmailAddress** from the list.

Now that you have successfully identified the control, the property of the control containing the desired value, and the variable in which to store the property value, the step text should read as follows:

```
Put the 'Text' property of the control 'Name=autoquote:e-mail' into
'strEmailAddress'
```

The **Local variable name** value in both the **Properties** pane and the step text changes to *strEmailAddress*.

To confirm that the test captures the property value and stores it properly, play back the test and review the result.

Playing Back and Analyzing the Enhanced Visual Test

Now that you have made several enhancements to the recorded test, play back the visual test and analyze the result.

1. Perform one of the following steps:
   - Choose **Actions ➤ Playback**.
   - Click **Playback** on the toolbar.

   The **Playback** dialog box opens.
2. In the **Result description** text box, type **`Enhanced test results for the recorded visual test`**.
3. Click **OK**.
   TestPartner plays back the enhanced test.
4. Click **Go to Result** on the **Playback Complete** dialog box.

   The **Result** window opens to the **Summary** tab by default.

   The **Summary** tab shows that the visual test passed, which means it played back successfully without any errors, or failed verifications.
5. Click the **Passed (1)** tab.

   The number in parentheses indicates the total number of verifications that passed. The **Test Steps** pane displays the verification step and the **Result Detail** column displays the pass text description of the verification.
6. Click the **Details** tab to display the result of every step.
7. In the **Test Steps** pane of the **Result** window, scroll down to the step that shows the result of storing the email address to a variable. The step text is similar to the following:

   ```
   Put the 'Text' property of the control 'Name=autoquote:e-mail' into
   'strEmailAddress'
   ```

8. The contents of the *strEmailAddress* variable displays in the **Result Detail** column. To view the entire contents of the **Text** property value, move your cursor over the **Result Detail** column for the step.

Congratulations! You have successfully created a visual test that reliably tests the sample application. In the next lesson, you will learn about several advanced testing concepts and features such as how to quickly and easily execute a visual test within another visual test.

Executing a Visual Test Within a Visual Test: Introduction

In this tutorial, you created a single visual test that performs every action required to receive an auto insurance quote from the web application. A single visual test is useful when implementing a basic test case against a simple application. However, most software testing requires a more rigorous approach that involves testing every aspect of an application. An additional requirement is the ability to rapidly update existing visual tests whenever the test application changes.

To provide an efficient means for solving these testing challenges, TestPartner supports modular testing, in which you can "chunk" common sets of actions of a particular testing solution into a single test, and then reuse the visual test in other visual tests that require the same set of actions.

Modular Testing

Before creating visual tests, test scripts, and other TestPartner assets to build application testing solutions, it is a good practice to plan a testing strategy. When creating testing solutions, it is not necessary to include all the parts of a particular test solution in a single visual test or test script, and is not usually beneficial to do so. When creating testing solutions, it is usually more efficient to approach your application testing in terms of a distinct series of transaction units. For example, testing an online ordering application might include the following distinct transaction units:

- Log on to the ordering application
- Create a customer profile
- Place orders
- Log off the ordering application

If a single test handles all of these distinct units, you would need a test of this type for each test scenario. For example, if any change occurs to the application such as an extra field is added to the logon window, then each test would require a change to accommodate data input to the new field.

Rather than creating one visual test or test script that tests all of these transaction units, and then recreating it ten times for each scenario, it may be more beneficial to create separate tests as test "modules" that handle each one of these transaction units. If a separate test is created for each of the separate transaction units and reused for each of the test scenarios, then only the test that handles the logon transaction unit would require change.

Now that you understand the basics of modular testing, you are ready to create a second test and add it to the test you created in the previous lessons.

Recording the Second Visual Test

In this section of the lesson, you record a second visual test for the tutorial and learn an alternate way to create a visual test asset.

1. Choose **File ➤ New**.
   The **New Asset** dialog box opens.
2. Select **Visual test** from the asset types list, and then type a name for the visual test asset in the **Asset name** text box.
   For this tutorial, type **`AddAccount`** for the name.
3. Check the **Begin Recording** check box to start recording immediately and then click **OK**.
4. From the **Home** page of the sample application, click **Sign Up** in the Login section.

The **Create A New Account** page opens.

5. Provide the following information in the appropriate fields. Press the **Tab** key to move from one field to the next.

Field Name	Value
First Name	Pat
Last Name	Smith
Birthday	February 12, 1990 **Note:** Click the down arrow next to the month and year in the calendar control to change the month and year and then select 12 on the calendar.
E-Mail Address	smith@test.com
Mailing Address	1212 Test Way
City	San Diego
State	CA
Postal Code	92121
Password	test

6. Click **Sign Up**.
7. Click **Continue**.
   The contact information is displayed.
8. Click **Home** near the top of the page to return to the home page where recording started.
9. Click **Log Out**.
10. Press **Alt+F10** to complete recording.
    The **Recording Complete** dialog box opens.
11. Click **Save**.
    The visual test opens in the Visual Navigator.

Inserting One Visual Test Within Another

In this section of the lesson, you will learn how to insert the second visual test, which adds a user account, in the original visual test before the steps that perform the request for an auto quote.

Executing visual tests within visual tests is a powerful method for efficiently testing the same basic steps in multiple visual tests.

Tip: When inserting a visual test within another visual test, it is important to ensure that any test applications are in the correct initial playback state.

1. From the **Recent** list in the **Start Screen**, click **AutoQuote** to open it.
   AutoQuote is the first visual test that you created in this tutorial.
2. In the **Test Steps** pane, select the <<Start>> step.
3. Click **Actions ➤ Insert ➤ Visual test**.
   The **Browse for Visual test** dialog box opens.
4. From the **Select an asset** list, select the visual test named **AddAccount** and then click **OK**.

TestPartner inserts a step before the selected step. The inserted step calls the selected visual test. The step text is as follows:

```
Playback visual test 'AddAccount'
```

Tip: During playback, when the preceding step executes, the inserted visual test plays back to completion before the original visual test plays back.

In the next lesson, you will learn how to playback this modular visual test, and respond to a playback error.

Responding to Playback Errors: Introduction

Errors encountered during playback can be caused by a variety of factors, such as changes in the test application and improper visual test step flow. Quickly diagnosing and fixing these errors using the debugging features minimizes visual test maintenance and allows for a more efficient team testing effort.

First, begin this lesson by playing back the modular test you created in the previous lesson.

Playing Back the Modular Test

In the previous lesson, you created a modular test by inserting one visual test, AddAccount, into another visual test, AutoQuote.

In this section of the lesson, you will play back the modular test and encounter an error during playback.

1. With the AutoQuote visual test open, click **Playback** on the toolbar.
   The **Playback** dialog box opens.
2. In the **Result description** text box, type **`Responding to errors in a modular test`**.
3. Click **OK**.

   During playback, the test stops on the **Create A New Account** page, red text displays, `Email is already registered`, and the following error message displays: `Playback error: Failed to find the attach name: "Name=signup:continue"`

   This error occurs because the database requires a unique email address for each customer record. Since you have already entered the email address during the recording of the AddAccount visual test, the email address already exists in the database and the test fails.

Now that you have encountered a playback error, you are ready to debug the test.

Debugging Errors

After TestPartner encounters a playback error, the TestPartner **Playback Error** dialog box opens and provides the option to enter Debug mode. In Debug mode, playback is suspended, which allows you to diagnose and fix any playback errors using the TestPartner debugging features.

In this section of the lesson, you will learn how to debug the error that occurred during playback of the modular test in the previous section.

1. From the **Playback Error** dialog box, click **Debug**.

   In Debug mode, playback is suspended. This allows you to fix the error by either editing the properties of the step incurring the error, deleting the step, disabling the step, or copying and pasting a step from another visual test before resuming playback.

TestPartner enters Debug mode and displays the AddAccount visual test with the step incurring the error highlighted in yellow.

2. Choose **Edit ➤ Enable/Disable** to disable the step `Click 'Name=signup:continue'`.

   By disabling this step, you prevent the error from occurring the next time you play back the test.

   The step text turns grey and italicized indicating that it is disabled.

3. Click **Playback** on the toolbar.
   The **Playback Error** dialog box opens.

4. Click **Debug**.

5. Choose **Edit ➤ Enable/Disable** to disable the following steps:

   - `Click 'Name=signup:continue'`
   - `Using 'InsuranceWeb Account Details - Windows Internet Explorer IEWindow'`
   - `Attach to 'Caption=InsuranceWeb: Account Details'`
   - `Click 'Name=logout-form:logout'`

   The error occurs because the test is looking for the **Log Out** button on the Home page, but since the Sign Up Continue step is disabled, the button does not display on the page. We will fix this error later in this tutorial.

6. Choose **Debug** from the menu.

   The following commands that appear at the top of the menu allow you to control the step execution:

Step Into (F8)	Executes playback one step at a time. **Step Into** is useful to trace through each step, and steps into other visual tests inserted into the visual test being played back. Each inserted visual test is also executed one step at a time. **Step Into** is useful for detailed analysis of a test, and lets you see the effect of each step or statement on variable usage and test application interaction.
Step Over (`Shift + F8`)	Executes an entire visual test inserted into another visual test as if it were a single step. Use **Step Over** when playback is in debug mode for a step that plays back a visual test. This plays back the inserted visual test in its entirety. Once the inserted visual test plays back in its entirety, playback suspends in debug mode at the next step in the original visual test. Using **Step Over** at a step other than one that plays back another visual test has the same effect as using **Step Into**. Only the next step executes before playback suspends and re-enters debug mode.
Step Out (`Ctrl + Shift + F8`)	Executes all remaining steps in a visual test being played back from another visual test, then suspends playback at the next step in the original visual test. Use **Step Out** when playback is suspended at a step in a visual test that has been inserted in another visual test, and you want to playback the remaining visual test and return to the original visual test. When playback executes the remainder of the inserted visual test, it suspends and re-enters debug mode at the next sequential line in the original visual test.
Run To Cursor (`Ctrl + F8`)	Allows you to select a step where you want playback suspended. This allows you to "step over" selected sections of a visual test. Use **Run To Cursor** to playback the visual test and stop playback at a point just before a run-time error occurs. This lets you stop playback at a specific line without having to insert breakpoints. Once playback stops, you can continue using one of the other debug options.

Run From Cursor Plays back the visual test from the currently selected test step.

7. Choose **Step Out**.

   This command executes the remaining steps in the AddAccount visual test, and then suspends playback in the next step of the AutoQuote visual test.

   After selecting **Step Out**, TestPartner displays the home page, which is the last step in AddAccount, and suspends playback. TestPartner then opens AutoQuote in Debug mode and highlights the next step.

Next, you learn how to monitor the values of variables used in the visual test.

Tracking Variables During Playback

In Debug mode, you can track the playback values of both local and reserved variables using the **Monitor Variables** window.

In this section of the lesson, you will learn how to use the **Monitor Variables** window to view the value of the local variable you created in a previous lesson.

1. Choose **Debug ➤ Monitor Variables**.
   The **Monitor Variables** window opens.
2. Expand the **Local and visual test variables** node.

   The local variable you created in a previous lesson, *strEmailAddress*, appears. During playback, the value for this variable will appear in the **Value** column once the step using the variable executes.

   Note: If you want to see the variable values in the **Monitor Variables** window but your test does not have any errors, insert a breakpoint in the visual test. To insert a breakpoint, select the step following the variable step, choose **Debug ➤ Set/Clear Breakpoint** and then playback the test.

3. To see how the **Run to Cursor** debugging works, select the step `Enter '2010'` and then choose **Debug ➤ Run to Cursor**.
   TestPartner plays back the remaining steps until the step `Enter '2010'`, and then displays the AutoQuote visual test. In the **Monitor Variables** window, review the value of the local variable.

Now you are ready to complete the playback of the modular test and review the test results.

Reviewing the Result

Review the results of the visual test after you have finished debugging the visual test.

1. Click **Playback** on the toolbar to complete playback of the AutoQuote visual test.
   The **Playback Complete** dialog box opens.
2. Click **Go to Result**.
   The AutoQuote result appears with the **Summary** tab displayed by default.

   The **Summary** tab displays the overall details of the test run. Note that the **Visual tests or .NET Scripts (number of times each ran)** field lists AutoQuote(1) and the inserted visual test, AddAccount(1).
3. Click the **Details** tab.
4. In the **Test Steps** pane, scroll down to the steps in blue text.

   By reviewing the **Result** and **Result Detail** columns, you can quickly find information about any errors that occurred during playback.

Note: To view all steps in the **Test Steps** pane, make sure you have Steps and Screen selected. Click **Actions ➤ View ➤ Steps and Screens**. The **Failed** tab does not display steps containing playback errors. It only displays failed checks and verifications.

Now that you have learned how to diagnose and debug playback errors, you are ready to modify the visual test to record additional steps to fix the error.

Modifying the Visual Test that Contains Errors

The errors within our visual test occur because the database requires a unique email address for each customer record. Since you have already entered the email address during the recording of the AddAccount visual test, the email address already exists in the database and the test fails. You can solve the database duplication error by recording additional steps within the test or you can add another modular test. For instance, we can record another visual test that resets the database and then insert the new test into the AddAccount visual test. Creating a modular test enables you to use the Reset Database steps within other visual tests as well.

1. Choose **File ➤ New**.
   The **New Asset** dialog box opens.
2. Select **Visual test** from the asset types list, and then type a name for the visual test asset in the **Asset name** text box.
   For this tutorial, type **`ResetDatabase`** for the name.
3. Check the **Begin Recording** check box to start recording immediately and then click **OK**.
4. From the **Home** page of the sample application, click **Settings** in the bottom right portion of the page.
5. In the **Reset Database** section, click **Reset**.
   This step is necessary to ensure that subsequent tests will not encounter errors when the duplicate data is entered.
6. Click **Home** near the top of the page to return to the home page where recording started.
7. Press **`Alt+F10`** to complete recording and then click **Save**.
8. Open the AddAccount visual test.
9. In the **Test Steps** pane, select the `<<Start>>` step.
10. Click **Actions ➤ Insert ➤ Visual test**.
    The **Browse for Visual test** dialog box opens.
11. From the **Select an asset** list, select the visual test named **ResetDatabase** and then click **OK**.

    TestPartner inserts a step before the selected step. The inserted step calls the selected visual test. The step text is as follows:

    ```
    Playback visual test 'ResetDatabase'
    ```

12. Enable all the steps that you previously disabled by selecting the steps and choosing **Edit ➤ Enable/Disable**.
13. Open the AutoQuote visual test and click **Playback** on the toolbar.
    All three visual tests complete without any errors.

Index

W