Rational Robot: Try it!

with PowerBuilder

WELCOME TO RATIONAL ROBOT

PowerBuilder!

Use the sample PowerBuilder applet with this *Try it!* to discover how easy it is to test the controls in your PowerBuilder applications. In minutes, you'll understand the power behind Rational Robot's Object Testing® as you record tests for these and other PowerBuilder controls:

- TreeViews and Tabs
- DataWindows
- Hidden DataWindows
- DropDownDataWindows
- DataStores
- ActiveX controls

If you're using PowerBuilderTM 5.0 or later to develop Windows NT® 4.0, Windows 2000, Windows® 98, or Windows® 95 applications, you need Rational® Robot — the leader in automated testing of client/server applications.

Robot lets you plan, develop, and execute tests for your Windows applications. Thanks to Robot's cross-Windows technology, tests created for one platform run smoothly on the others.

As you use the testing tips in this card on the PowerBuilder sample applet, you'll quickly discover how easy it is to record tests using Robot.

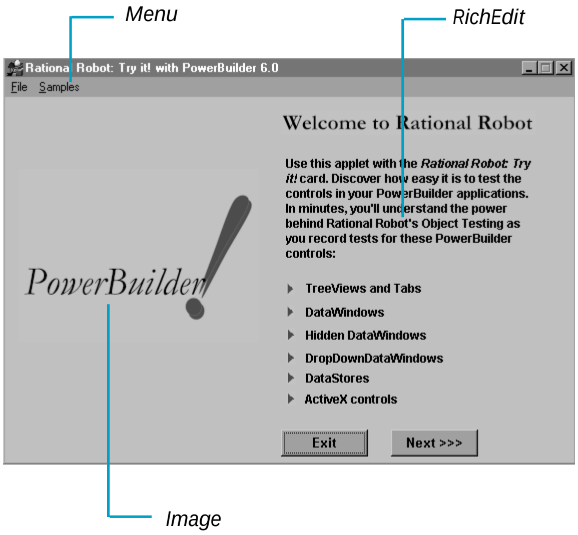
To find out more about Robot, be sure to take a look at the tutorial in *Getting Started with Rational Robot* and the online Help.

▸ ▸ ▸ Find out how Rational puts quality to the test. ▸ ▸ ▸

Rational®
the **e-development** company™

Test these controls in the sample applet

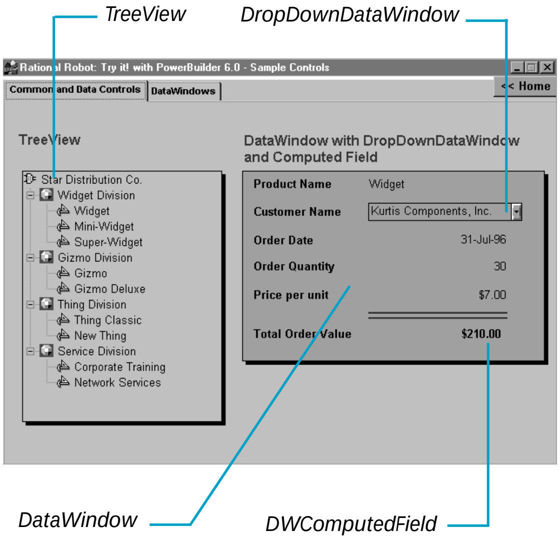
These three windows make up the **Rational Robot: Try it*!* with PowerBuilder** applet. Each window contains numerous controls to test. Read on to find out how. ► ► ►



Welcome to Rational Robot

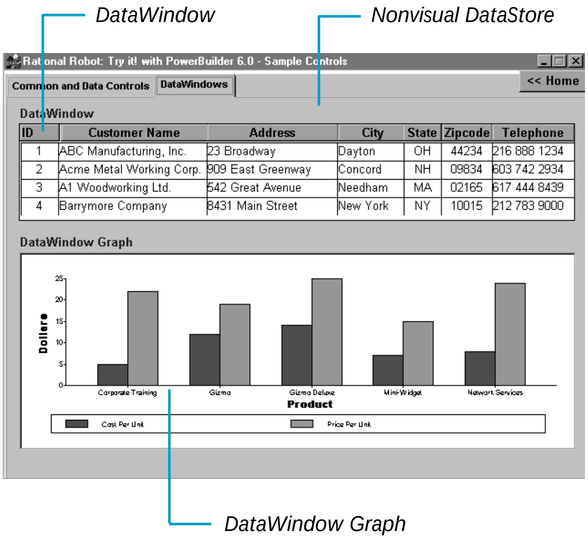
Use this applet with the *Rational Robot: Try it!* card. Discover how easy it is to test the controls in your PowerBuilder applications. In minutes, you'll understand the power behind Rational Robot's Object Testing as you record tests for these PowerBuilder controls:

- TreeViews and Tabs
- DataWindows
- Hidden DataWindows
- DropDownDataWindows
- DataStores
- ActiveX controls



DataWindow with DropDownDataWindow and Computed Field

Product Name	Widget
Customer Name	Kurtis Components, Inc.
Order Date	31-Jul-96
Order Quantity	30
Price per unit	$7.00
Total Order Value	$210.00



DataWindow

ID	Customer Name	Address	City	State	Zipcode	Telephone
1	ABC Manufacturing, Inc.	23 Broadway	Dayton	OH	44234	216 888 1234
2	Acme Metal Working Corp.	909 East Greenway	Concord	NH	09834	603 742 2934
3	A1 Woodworking Ltd.	542 Great Avenue	Needham	MA	02165	617 444 8439
4	Barrymore Company	8431 Main Street	New York	NY	10015	212 783 9000

Start the applications and log in to the repository

Before you begin to test, you'll install the PowerBuilder sample applet. When you install the applet, a sample repository is created automatically. A repository contains all the information about your testing project, including test documents, requirements, scripts, logs, defects, queries, and reports.

Then, you'll start the PowerBuilder sample applet and Rational Robot**.** You'll use Robot to record both your actions as you navigate through the sample applet and the verification points that you insert to test the contents of specific controls.

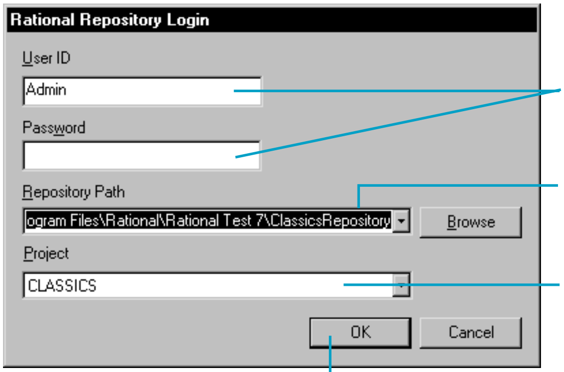
► ► ► ***Try this:***

❶ Click **Start** → **Programs** → ***Rational product name*** → **Rational Test** → **Setup Rational Test Samples**.

This installs the sample PowerBuilder applet and creates a sample repository called **ClassicsRepository** and a sample project called **Classics**.

❷ Click **Start** → **Programs** → **Rational Test Samples** → **PowerBuilder**.

❸ Click **Start** → **Programs** → ***Rational product name*** → **Rational Test** → **Rational Robot** to open the Rational Repository Login dialog box**.**



User ID: Admin

Password:

Repository Path: ogram Files\Rational\Rational Test 7\ClassicsRepository

Project: CLASSICS

*Type your **User ID** and **Password**. If you do not know these, see your administrator.*

*Select the sample repository, **ClassicsRepository**.*

*Select the project name **Classics**.*

*Click **OK** to log in.*

► ► ► For more about Rational repositories, see the *Using the Rational Administrator* manual. ► ► ►

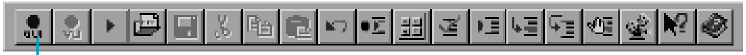
Start recording and test a control’s properties

When you record, you’ll use Robot’s **Object-Oriented Recording**â to test controls, independent of their positions in the GUI. If controls change locations or their text changes, Robot will still find them on playback because Robot identifies controls by their PowerBuilder object names.

Robot offers many verification points for use during recording. (See the list on the last page.) One of the most powerful — **Object Properties** — lets you capture the properties of controls, even hidden ones such as DataWindows and DataStores. Robot’s easy-to-use **Object Testing**â lets you inspect and verify all of a control’s properties.

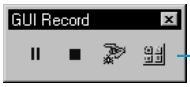
► ► ► ***Try this:***

❶



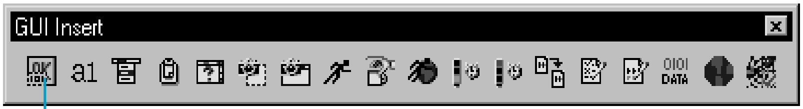
*On the Robot toolbar, click to start recording. Type a name for the script, such as **PBtest**, and click **OK**.*

❷



*Click to display the **GUI Insert** toolbar.*

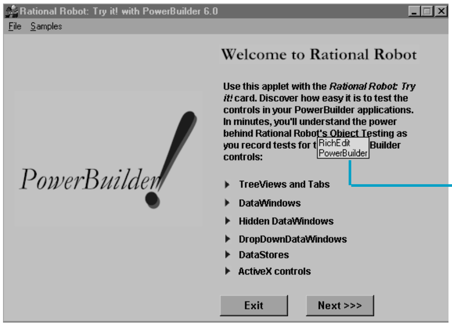
❸



*For the first test, click the **Object Properties** button. A dialog box appears. Type a name, such as **Test1**, and click **OK**.*

❹ In the Select Object dialog box, click the left mouse button on the Object Finder tool and keep the mouse button depressed. The dialog box disappears.

❺



Welcome to Rational Robot

Use this applet with the *Rational Robot: Try it!* card. Discover how easy it is to test the controls in your PowerBuilder applications. In minutes, you'll understand the power behind Rational Robot's Object Testing as you record tests for t Builder controls:

- TreeViews and Tabs
- DataWindows
- Hidden DataWindows
- DropDownDataWindows
- DataStores
- ActiveX controls

*Keeping the mouse button depressed, move the Object Finder tool around the window and look at the TestTip that describes each control. Point to the RichEdit text and release the left mouse button. Click **OK** in the Select Object dialog box.*

*Notice how Robot captures all of a control’s properties. Scroll through the list of properties and click **OK** to complete the Object Properties test. Then go to the next page to try some more tests.*

Test data in PowerBuilder controls

The **Object Data** verification point is another of Robot's comprehensive verification points. It lets you capture data from visible and hidden controls, including standard Windows controls, ActiveX controls, lists, menus, DataWindows, DropDownDataWindows, and DataStores.

► ► ► *Try this:*

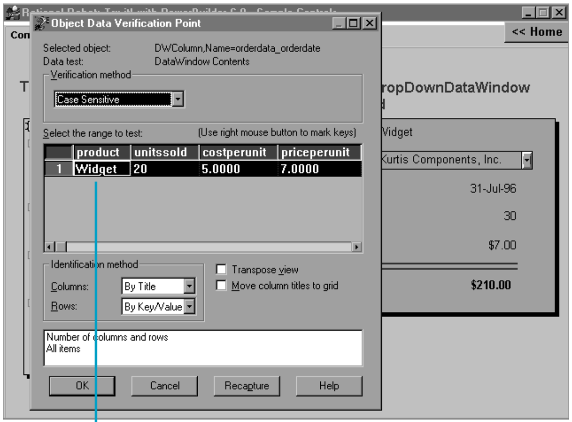
❶ Click **Next** to display the **Common and Data Controls** tab.

❷ Insert an Object Data verification point.

❸ Type the name **Test2** and click **OK**.

❹ Use the left mouse button to drag the Object Finder tool onto the **DWColumn** to the right of the **Order Date** and release the mouse button.

❺ Click **OK** twice to complete the test.



Object Data Verification Point

Selected object: DWColumn,Name=orderdata_orderdate
Data test: DataWindow Contents

Verification method: Case Sensitive

Select the range to test: (Use right mouse button to mark keys)

	product	unitssold	costperunit	priceperunit
1	Widget	20	5.0000	7.0000

Identification method
Columns: By Title
Rows: By Key/Value

□ Transpose view
□ Move column titles to grid

Number of columns and rows
All items

The captured data includes all of the data in the record. You can select the data to test, and you can change verification and identification methods.

► ► ► *Try this:*

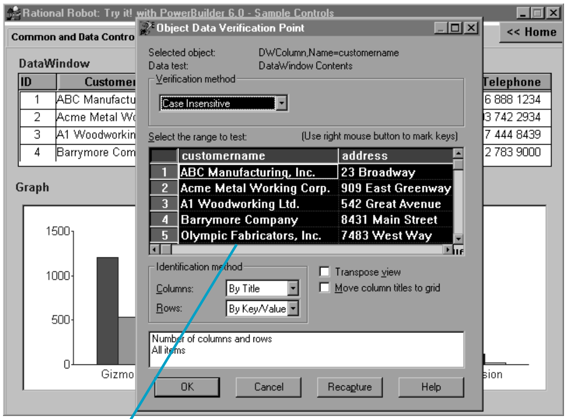
❶ Click the **DataWindows** tab.

❷ Insert an Object Data verification point.

❸ Type the name **Test3** and click **OK**.

❹ Drag the Object Finder tool onto a name in the Customer Name column and release the mouse button.

❺ Click **OK** twice to complete the test.



Rational Robot: Try it! with PowerBuilder 6.0 - Sample Controls

Object Data Verification Point

Selected object: DWColumn,Name=customername
Data test: DataWindow Contents

Verification method: Case Insensitive

Select the range to test: (Use right mouse button to mark keys)

	customername	address
1	ABC Manufacturing, Inc.	23 Broadway
2	Acme Metal Working Corp.	909 East Greenway
3	A1 Woodworking Ltd.	542 Great Avenue
4	Barrymore Company	8431 Main Street
5	Olympic Fabricators, Inc.	7483 West Way

Identification method
Columns: By Title
Rows: By Key/Value

□ Transpose view
□ Move column titles to grid

Number of columns and rows
All items

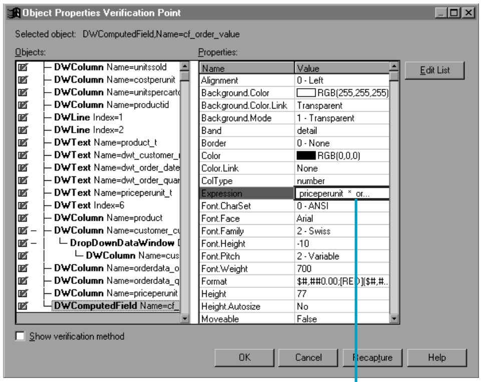
The captured data includes data not displayed onscreen in the DataWindow.

Test a computed field and a DataWindow graph

To test a computed field in the sample applet, you'll use the **Object Properties** verification point, and to test the data in the DataWindow Graph you'll use the **Object Data** verification point. The Object Properties verification point lets you capture the properties of *any* control in an application, and the Object Data verification point lets you capture the data in many databound controls. And, if for some reason, an object data test does not exist for an object, you can create your own data test. For instructions, see the *Using Rational Robot* manual.

▶ ▶ ▶ ***Try this:***

1. Click the **Common and Data Controls** tab.
2. Insert an Object Properties verification point.
3. Type the name **Test4** and click **OK**.
4. Drag the Object Finder tool onto the **DataWindow** and release the mouse button. Click **OK** to capture the 21 objects it contains.
5. Click **DWComputedField**, the last object in the Objects List, to display its properties.
6. Under the **Value** column, double-click the **Expression** value **priceperunit...** to display the computed field's expression, and click **OK**.
7. Click **OK** to complete the test.



Selected object: DWComputedField,Name=cf_order_value

Objects:

☑ ├ **DWColumn** Name=unitssold
☑ ├ **DWColumn** Name=costperunit
☑ ├ **DWColumn** Name=unitspercarto
☑ ├ **DWColumn** Name=productid
☑ ├ **DWLine** Index=1
☑ ├ **DWLine** Index=2
☑ ├ **DWText** Name=product_t
☑ ├ **DWText** Name=dwt_customer_
☑ ├ **DWText** Name=dwt_order_date
☑ ├ **DWText** Name=dwt_order_quar
☑ ├ **DWText** Name=priceperunit_t
☑ ├ **DWText** Index=6
☑ ├ **DWColumn** Name=product
☑ – ├ **DWColumn** Name=customer_cu
☑ – │ └ **DropDownDataWindow** [
☑ │ └ **DWColumn** Name=cus
☑ ├ **DWColumn** Name=orderdata_o
☑ ├ **DWColumn** Name=orderdata_q
☑ ├ **DWColumn** Name=priceperunit
☑ └ **DWComputedField** Name=cf_

Properties:

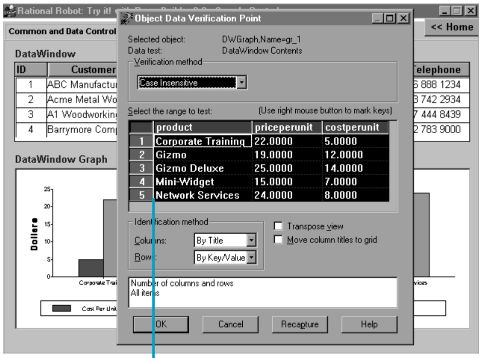
Name	Value
Alignment	0 - Left
Background.Color	□ RGB(255,255,255)
Background.Color.Link	Transparent
Background.Mode	1 - Transparent
Band	detail
Border	0 - None
Color	■ RGB(0,0,0)
Color.Link	None
ColType	number
Expression	priceperunit * or...
Font.CharSet	0 - ANSI
Font.Face	Arial
Font.Family	2 - Swiss
Font.Height	-10
Font.Pitch	2 - Variable
Font.Weight	700
Format	$#,##0.00;[RED]($#,#..
Height	77
Height.Autosize	No
Moveable	False

□ Show verification method

The Object Properties test captures the expression for the computed field.

▶ ▶ ▶ ***Try this:***

1. Click the **DataWindows** tab.
2. Insert an Object Data verification point.
3. Type the name **Test5** and click **OK**.
4. Drag the Object Finder tool onto the **DataWindow Graph** and release the mouse button.
5. Click **OK** to capture all the data represented in the graph.
6. Click **OK** to complete the test.



Selected object: DWGraph,Name=gr_1
Data test: DataWindow Contents

Verification method: Case Insensitive

Select the range to test: (Use right mouse button to mark keys)

	product	priceperunit	costperunit
1	Corporate Training	22.0000	5.0000
2	Gizmo	19.0000	12.0000
3	Gizmo Deluxe	25.0000	14.0000
4	Mini-Widget	15.0000	7.0000
5	Network Services	24.0000	8.0000

Identification method
Columns: By Title
Rows: By Key/Value

□ Transpose view
□ Move column titles to grid

Number of columns and rows
All items

The test captures all of the data represented in the graph.

Test nonvisual controls

Many PowerBuilder applications have hidden controls that you can’t test manually because you can’t see them. Fortunately, Rational Robot sees all, letting you test any hidden or nonvisual control that resides in the application or on the desktop. Here’s an easy way to display and test these controls.

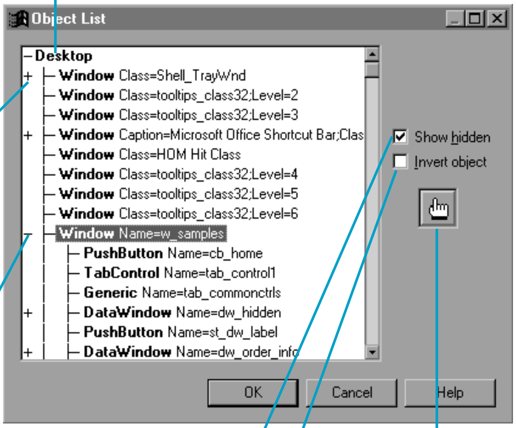
▶ ▶ ▶ ***Try this:***

❶ Click the **Common and Data Controls** tab.

❷ Insert an Object Properties verification point.

❸ Type the name **Test6** and click **OK**.

❹ Click the **Browse** button to display the Object List.



Object List

```
–Desktop
+ ├Window Class=Shell_TrayWnd
  ├Window Class=tooltips_class32;Level=2
  ├Window Class=tooltips_class32;Level=3
+ ├Window Caption=Microsoft Office Shortcut Bar;Clas
  ├Window Class=HOM Hit Class
  ├Window Class=tooltips_class32;Level=4
  ├Window Class=tooltips_class32;Level=5
  ├Window Class=tooltips_class32;Level=6
– ├Window Name=w_samples
  │ ├PushButton Name=cb_home
  │ ├TabControl Name=tab_control1
  │ ├Generic Name=tab_commonctrls
+ │ ├DataWindow Name=dw_hidden
  │ ├PushButton Name=st_dw_label
+ │ ├DataWindow Name=dw_order_info
```

☑ Show hidden

□ Invert object

The top of the tree is the desktop.

Top-level objects that contain other objects are preceded by a plus sign.

Double-click top-level objects to expand them.

When selected, shows hidden objects in the list.

When selected, select an object in the list and see its colors invert in the applet.

When clicked, returns to selection with the Object Finder tool.

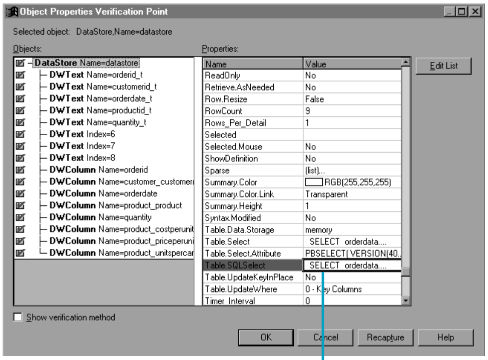
❺ Make sure the **Show hidden** option is selected.

❻ Scroll down through the objects until you see **DataStore — a** hidden data control.

❼ Select **DataStore** and click **OK** to display its properties.

❽ Scroll till you see **Table.SQLSelect** under the **Name** column. Double-click **SELECT orderdata...** to display the entire SQL statement and click **OK**.

❾ Click **OK** to complete the test.



Object Properties Verification Point

Selected object: DataStore,Name=datastore

Objects:

```
–DataStore Name=datastore
  ├DWText Name=orderid_t
  ├DWText Name=customerid_t
  ├DWText Name=orderdate_t
  ├DWText Name=productid_t
  ├DWText Name=quantity_t
  ├DWText Index=6
  ├DWText Index=7
  ├DWText Index=8
  ├DWColumn Name=orderid
  ├DWColumn Name=customer_customer
  ├DWColumn Name=orderdate
  ├DWColumn Name=product_product
  ├DWColumn Name=quantity
  ├DWColumn Name=product_costperunit
  ├DWColumn Name=product_priceperuni
  └DWColumn Name=product_unitspercar
```

Properties:

Name	Value
ReadOnly	No
Retrieve.AsNeeded	No
Row.Resize	False
RowCount	9
Rows_Per_Detail	1
Selected	
Selected.Mouse	No
ShowDefinition	No
Sparse	(list)...
Summary.Color	□ RGB(255,255,255)
Summary.Color.Link	Transparent
Summary.Height	1
Syntax.Modified	No
Table.Data.Storage	memory
Table.Select	SELECT orderdata....
Table.Select.Attribute	PBSELECT(VERSION(40..
Table.SQLSelect	SELECT orderdata....
Table.UpdateKeyInPlace	No
Table.UpdateWhere	0 - Key Columns
Timer_Interval	0

□ Show verification method

Notice that an SQL statement is the value of one of the properties. Double-click to see the entire statement.

What's next

You've just tested some of the controls in the PowerBuilder sample applet. To record more tests, try a Menu verification point on the first window, an Alphanumeric verification point on text controls, and an Object Properties verification point on any of the controls you haven't tested.

When you're done, stop recording by pressing this button on the **GUI Record** toolbar: ■

The value of verification points becomes even more apparent when you play them back. During playback, verification points identify changes or unintentional errors in an application as it evolves. This lets you correct any errors *before* you deliver the application to customers.

For playback instructions, see the *Using Rational Robot* manual. Before you play back the scripts recorded against the sample applet, return to the first window in the applet by clicking **Home**. If any verification points fail on playback, just double-click the failure in the Rational LogViewer to see why.

To play back a script, press this button on the Robot toolbar: ▶

For valuable information about testing PowerBuilder applications, see the *Testing PowerBuilder Applications* chapter in the *Using Rational Robot* manual.

Rational Robot verification points

Use these verification points to ensure delivery of high-quality applications to your customers.

Object Properties — Captures and tests properties of visible and hidden controls.

Object Data — Captures and tests data from visible and hidden controls.

Alphanumeric — Captures and tests alphanumeric data in objects that contain text.

Clipboard — Captures and tests alphanumeric data copied to the Clipboard.

Web Site Scan — Checks the contents of a Web site with every revision and provides a report on defects.

Menu — Captures and tests text, shortcut keys, and the state of menus in as many as five levels of submenus.

Window Image — Captures a window as a bitmap image.

Region Image — Captures a specific region of the screen as a bitmap image.

Window Existence — Verifies the existence and status of a specific window or dialog box.

Web Site Compare — Captures a baseline of a Web site and compares it to the Web site at another point in time.

▶ ▶ ▶ For more about verification points, see the *Using Rational Robot* manual. ▶ ▶ ▶