

WELCOME TO RATIONAL ROBOT

Visual
Basic



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

- ▶ Microsoft Chart control
- ▶ Data Bound List control
- ▶ MS FlexGrid control
- ▶ Microsoft Windows common controls
- ▶ Rich Text Box control

If you're using Visual Basic® 4.0 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.

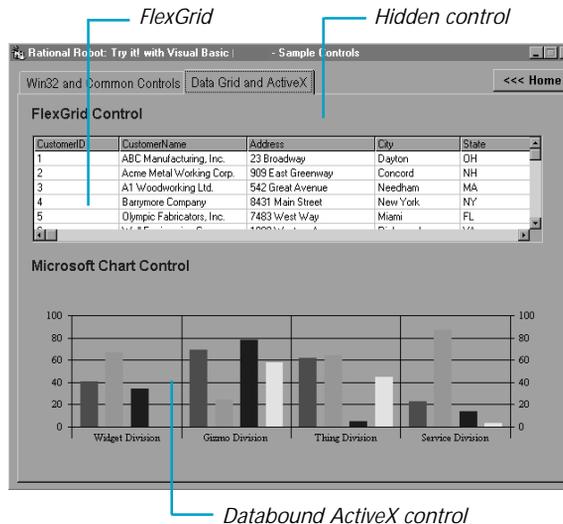
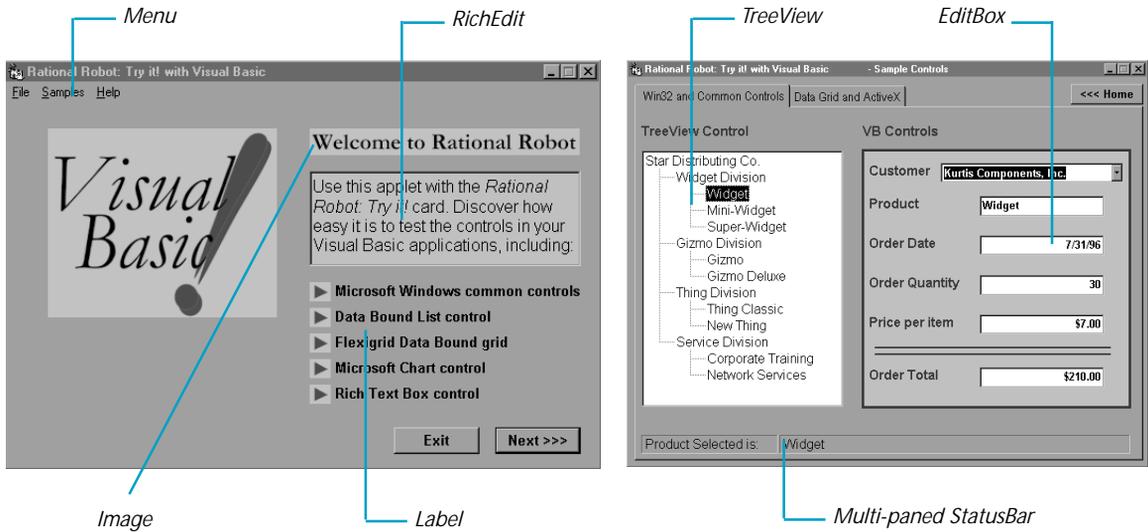
As you use the testing tips in this card on the Visual Basic sample applet, you'll quickly discover how easy it is to record tests using Robot.

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. ▶ ▶ ▶

Test these controls in the sample applet

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



Start the applications and log in to the repository

Before you begin to test, you'll install the Visual Basic 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 Visual Basic 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 verify the contents of specific controls.

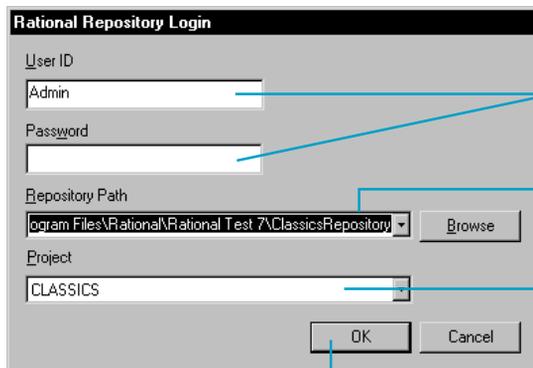
▶ ▶ ▶ *Try this:*

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

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

- 2 Click **Start** → **Programs** → **Rational Test Samples** → **Visual Basic**.

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



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^a** 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 Visual Basic 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 *any* control, even a hidden one. Robot's easy-to-use **Object Testing^a** lets you inspect and verify all of a control's properties.

▶ ▶ ▶ Try this:



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



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



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

- 4 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.



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** control 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 Visual Basic controls

The **Object Data** verification point is another of Robot's comprehensive verification points. It lets you capture data from both visible and hidden controls, including standard Windows controls, ActiveX controls, lists, menus, and data controls. Feel free to repeat these tests as often as you want.

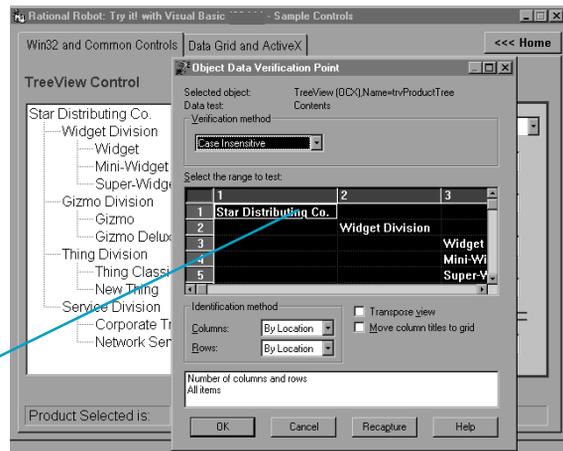
▶ ▶ ▶ **Try this:**

- 1 Click **Next** to display the **Win32 and Common Controls** tab.
- 2 Insert an Object Data verification point.



- 3 Type the name **Test2** and click **OK**.
- 4 Use the left mouse button to drag the Object Finder tool onto the TreeView and release the mouse button.
- 5 Click **OK** twice to complete the test.

The captured data includes each branch in the tree. You can select the data to test, and change verification and identification methods.

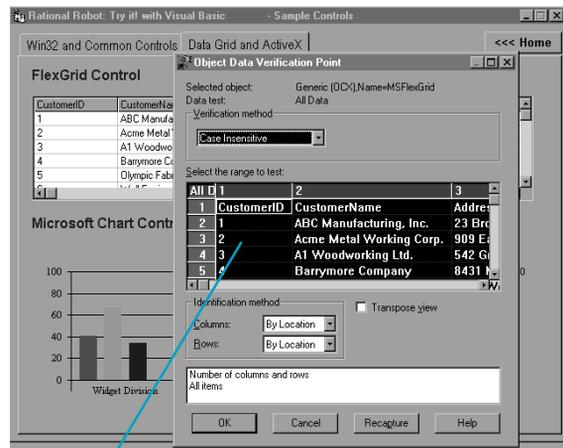


▶ ▶ ▶ **Try this:**

- 1 Click the **Data Grid and ActiveX** tab.
- 2 Insert an Object Data verification point.
- 3 Type the name **Test3** and click **OK**.
- 4 Drag the Object Finder tool onto a row in the FlexGrid control and release the mouse button.
- 5 Click **OK** twice to accept the **All Data** test.
- 6 Click **OK** to complete the test.



The captured data includes all the data displayed in the FlexGrid control.



Test ActiveX controls

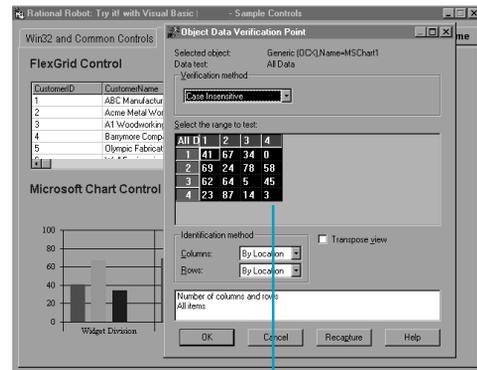
To test the ActiveX controls in the sample applet, you'll use the **Object Data** and **Object Properties** verification points. The Object Properties verification point lets you capture the properties of *any* control in an application. And, if for some reason, an object data test does not exist for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

▶ ▶ ▶ Try this:

- 1 Click the **DataGrid and ActiveX** tab.
- 2 Insert an Object Data verification point.

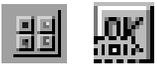


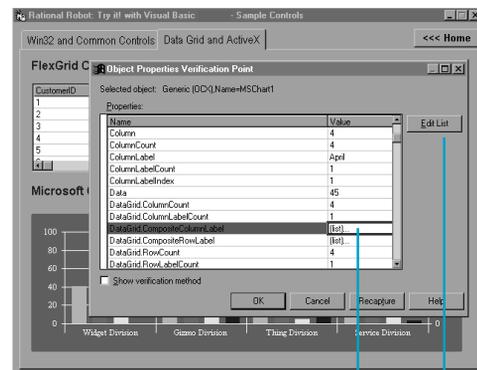
- 3 Type the name **Test4** and click **OK**.
- 4 Drag the Object Finder tool onto the Chart control and release the mouse button.
- 5 Click **OK** twice to accept the **All Data** test.
- 6 Click **OK** to complete the test.



The **All Data** test replaces graphical bars with their numeric values.

▶ ▶ ▶ Try this:

- 1 Click the **DataGrid and ActiveX** tab.
 - 2 Insert an Object Properties verification point.
- 
- 3 Type the name **Test5** and click **OK**.
 - 4 Drag the Object Finder tool onto the Chart control and release the mouse button. Click **OK**.
 - 5 Scroll through the object properties of the chart until you see the **DataGrid.CompositeRowLabel** property.
 - 6 Under the **Value** column, double-click **list...** to view the Property dialog box.
 - 7 Click **OK** twice to complete the test.



Double-clicking a value such as **list...** opens the Property dialog box, so you can edit that property.

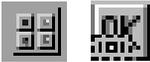
The **Edit List** button lets you add or remove properties from the verification point.

Test hidden controls

Many Visual Basic 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 control that resides in the application or on the desktop. Here's an easy way to display and test these controls.

▶ ▶ ▶ Try this:

- 1 Click the **DataGrid and ActiveX** tab.
- 2 Insert an Object Properties verification point.



- 3 Type the name **Test6** and click **OK**.
- 4 Click the **Browse** button to display the Object List.

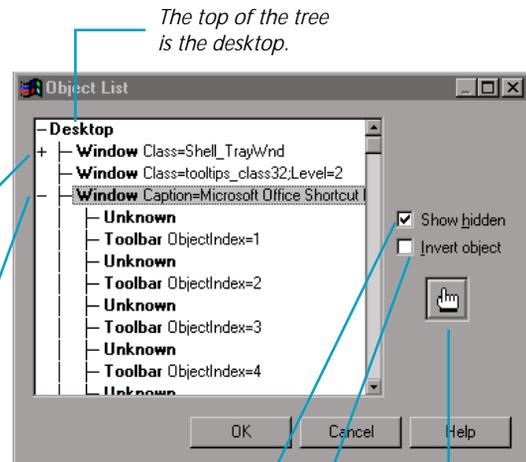
Top-level objects that contain other objects are preceded by a plus sign.

Double-click top-level objects to expand them.

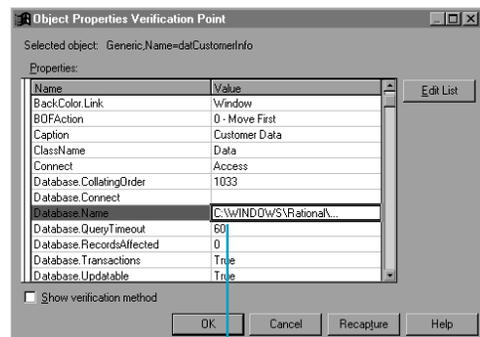
When selected, shows hidden objects in the list.

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.



- 5 Make sure the **Show hidden** option is selected. Scroll to locate **Window Name=frmSamples** and make sure it is selected and its subobjects displayed.
- 6 Scroll through the subobjects until you see **Generic Name=datCustomerInfo** — a hidden data control.
- 7 Select it and click **OK** to display its properties.
- 8 Under the **Value** column, double-click the **DatabaseName** value. Notice the entire path and name is displayed. Click **OK**.
- 9 Click **OK** to complete the test.



Notice that the database path and name is the value of one of the properties.

What's next

You've just tested some of the controls in the Visual Basic 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 Visual Basic applications, see the *Testing Visual Basic 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.		Menu — Captures and tests text, shortcut keys, and the state of menus in as many as five levels of submenus.
	Object Data — Captures and tests data from visible and hidden controls.		Window Image — Captures a window as a bitmap image.
	Alphanumeric — Captures and tests alphanumeric data in objects that contain text.		Region Image — Captures a specific region of the screen as a bitmap image.
	Clipboard — Captures and tests alphanumeric data copied to the Clipboard.		Window Existence — Verifies the existence and status of a specific window or dialog box.
	Web Site Scan — Checks the contents of a Web site with every revision and provides a report on defects.		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. ▶ ▶ ▶