

## WELCOME TO RATIONAL ROBOT



Use the sample Java applet with this *Try it!* to discover how easy it is to test the objects in your Java programs. In minutes, you'll understand the power behind Rational Robot's Object Testing<sup>®</sup> as you record tests for these and other Java JFC objects:

- ▶ JTree
- ▶ JTable
- ▶ JComboBox
- ▶ JSplitPane
- ▶ JTextArea
- ▶ JRadioButton

If you develop Java™ programs, you need Rational® Robot — the leader in automated testing of GUI applications.

Robot lets you plan, develop, and execute tests for your Java applets and applications. Robot supports multiple Java environments hosted on Microsoft® Windows®, including Sun® JDK™ and JRE, Netscape® Navigator, and Microsoft Internet Explorer and Jview. Thanks to Robot's cross-environment technology, tests created for one environment run smoothly on the others.

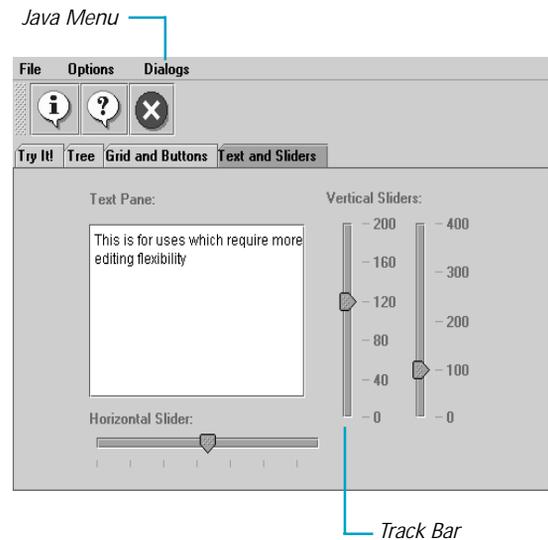
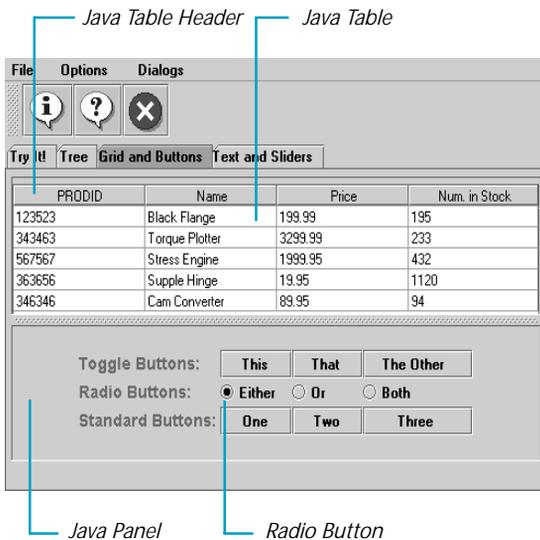
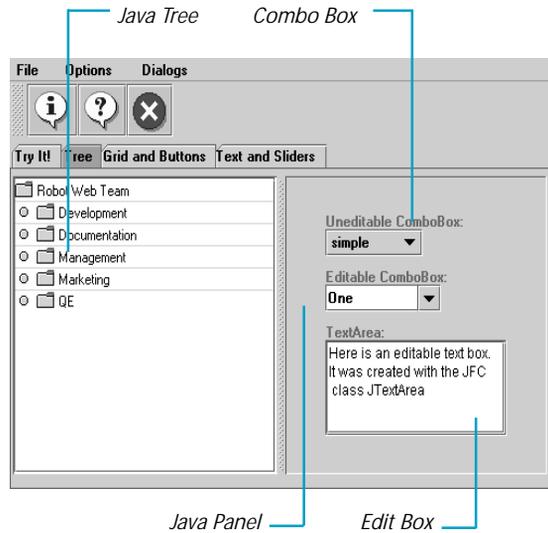
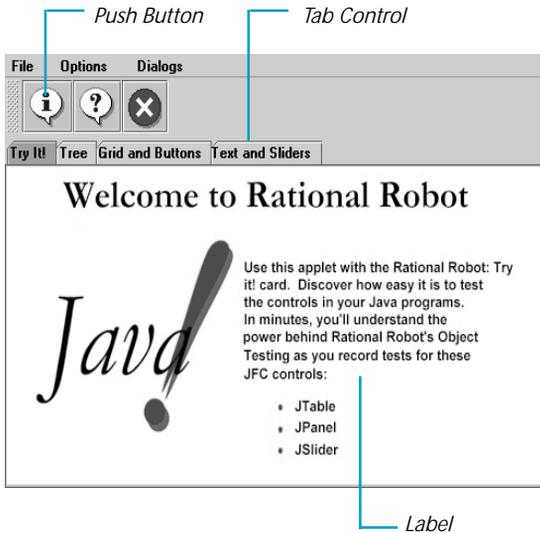
With Robot, you can test Java applications created with a variety of Java development tools and GUI class libraries. As you use the testing tips in this card with the sample Sun Swing/JFC applet, you'll quickly discover how easy it is to record tests using Robot.

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 objects in the sample applet

These four windows make up the **Rational Robot: Try it! with Java** applet. Each window contains numerous objects to test. Read on to find out how. ▶ ▶ ▶

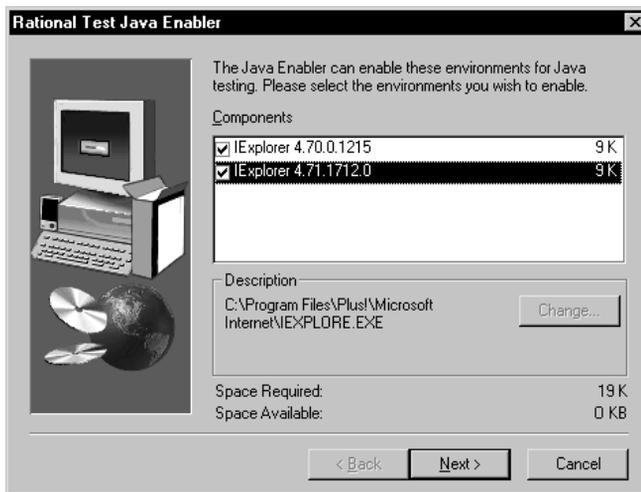


## Run the Java Enabler

Before you begin to test, you must run the Java Enabler. The **Java Enabler** is a wizard that makes each supported host environment testable. Host environments include Web browsers, applet viewers, JDKs, and JVMs. The Java Enabler also contains an option that takes you directly to the Rational Web site where you can download additional Java support.

### ▶ ▶ ▶ *Try this:*

- 1 Click **Start** → **Programs** → *Rational product name* → **Rational Test** → **Java Enabler**.
- 2 When the Enabler asks if you want to continue, click **Yes**.  
The Enabler searches for the environments on your system.
- 3 Select the environments that you want to enable.



- 4 Click **Next**.
- 5 Click **Yes** to view the log file.

▶ ▶ ▶ For more about the Java Enabler, see the *Using Rational Robot* manual. ▶ ▶ ▶

## Start the applications and log in to the repository

After you run the Java Enabler, you'll install the sample Java 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 Java applet and start 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 specific objects.

### ▶ ▶ ▶ *Try this:*

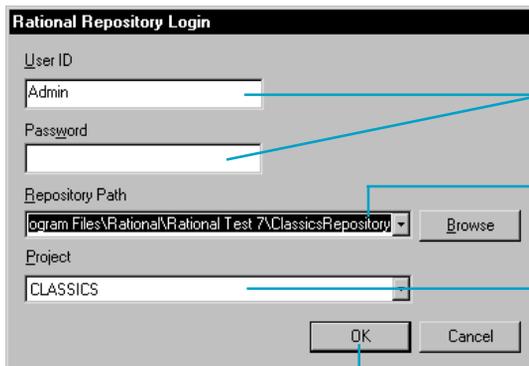
- 1 Click **Start** → **Programs** → *Rational product name* → **Rational Test** → **Setup Rational Test Samples**.

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

- 2 Click **Start** → **Programs** → **Rational Test Samples** → **Java**.

Note: The sample Java applet requires Sun's Swing foundation classes version 1.03 or later. For information about installing Swing, see the chapter *Testing Java Applets and Applications* in the *Using Rational Robot* manual.

- 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 an object's properties

When you record, you'll use Robot's **Object-Oriented Recording**<sup>®</sup> to test objects, independent of their positions in the GUI. If objects change locations or their text changes, Robot will still find them on playback because Robot can identify objects by their internal Java names, or by text or index.

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* object. Robot's easy-to-use **Object Testing**<sup>®</sup> lets you inspect and verify all of an object's properties.

## ▶ ▶ ▶ Try this:



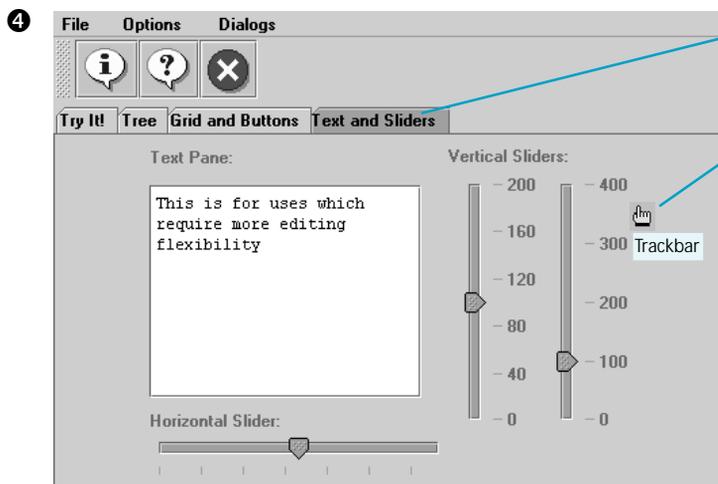
On the Robot toolbar, click to start recording. Type a name for the script, such as *Java1*, and click *OK*.



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*.



Click the *Text and Sliders* tab.

Then, drag the *Object Finder* tool around the window and look at the *TestTips* that describe each object. Point to one of the *Trackbar* objects, release the left mouse button, and click *OK*.

Notice how Robot captures all of an object's properties. Scroll through the list of properties and click *OK* to complete the test. Then, go to the next page to try some more tests.

# Test data in Java objects

The **Object Data** verification point is another of Robot's comprehensive verification points. It lets you capture data from any Java object, including labels, buttons, tab controls, menus, combo and edit boxes, trees, grids, panels, trackbars, and more.

## ▶ ▶ ▶ Try this:

- 1 Click the **Tree** tab.
- 2 Insert an Object Data verification point.



- 3 Type a name, such as **Test2**, and click **OK**.
- 4 Drag the Object Finder tool to the Editable ComboBox until **ComboBox** appears in the TestTip.

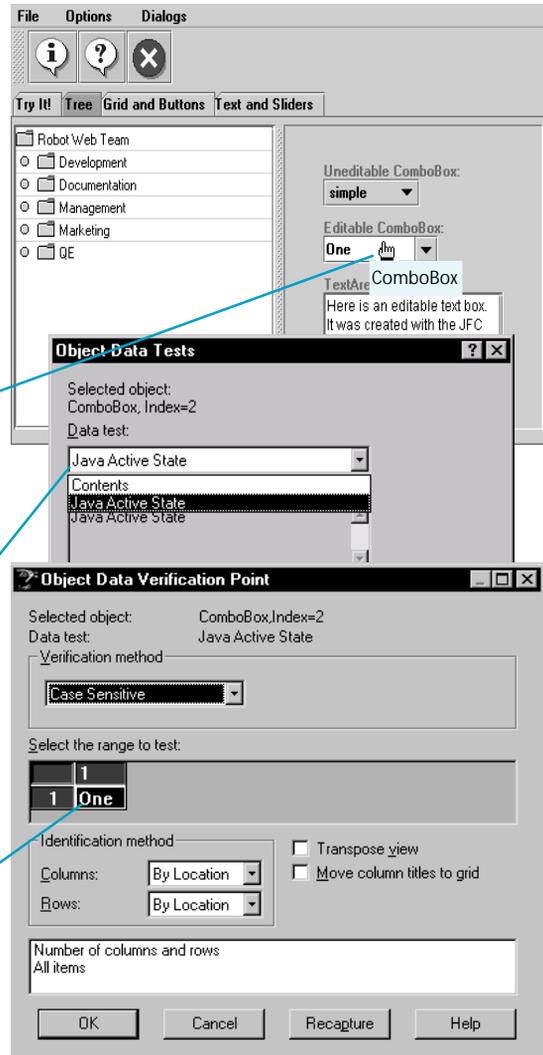
*Object Finder tool*

- 5 Release the mouse button and click **OK**.
- 6 In the Object Data Tests dialog box, select **JavaActiveState**.

*This data test lets you verify that a particular element in the list is selected. (The **Contents** data test, on the other hand, captures all of the items in the list.)*

- 7 Click **OK**.
- 8 View the captured data and click **OK**.

*The captured data includes the selected item from the test.*



# Test the contents of a Java panel

A feature unique to Java testing is the ability to collect and test the data for all the known objects on a Java panel. A **panel** is a container of objects and other panels that you have grouped together. To test the contents of a Java panel in the sample applet, you'll use the **Object Data** verification point.

## ▶ ▶ ▶ Try this:

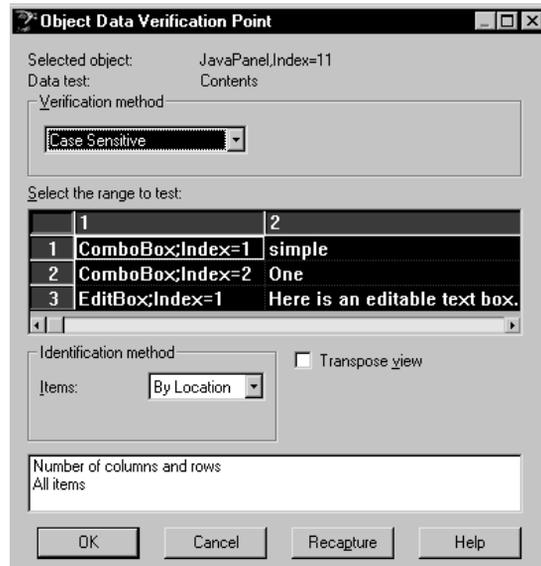
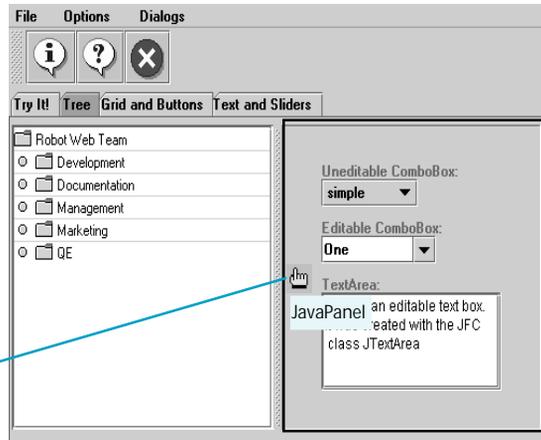
- 1 From the Tree tab, insert an Object Data verification point.



- 2 Type a name, such as **Test3**, and click **OK**.
- 3 Drag the Object Finder tool to the right side of the window until **JavaPanel** appears in the TestTip.

*This panel includes two combo boxes and a text area.*

- 4 Release the mouse button and click **OK**.
- 5 View the captured data and click **OK** to complete the test.



## What's next

You've just tested some of the objects in the sample Java applet. To record more tests, try an Object Data verification point on the JavaTree, and another on the Grid.

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 scripts recorded against the sample applet, return to the first window in the applet by clicking the **TryIt!** tab. 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 Java applications, see the *Testing Java Applets and Applications* chapter in the *Using Rational Robot* manual.

## Rational Robot verification points

Use the Object Properties and Object Data verification points to ensure delivery of high-quality Java applications to your customers. For a brief description of these and other verification points used in Robot, see the table below.

	<b>Object Properties</b> – Captures and tests properties of visible and hidden objects.		<b>Menu</b> – Captures and tests text, shortcut keys, and the state of menus in as many as five levels of submenus.
	<b>Object Data</b> – Captures and tests data from visible and hidden objects.		<b>Window Image</b> – Captures a window as a bitmap image.
	<b>Alphanumeric</b> – Captures and tests alphanumeric data in windows that contain text.		<b>Region Image</b> – Captures a specific region of the screen as a bitmap.
	<b>Clipboard</b> – Captures and tests alphanumeric data copied to the Clipboard.		<b>Window Existence</b> – Verifies the existence of a specific window or dialog box.
	<b>Web Site Scan</b> – Checks the contents of a Web site with every revision and provides a report on defects.		<b>Web Site Compare</b> – 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. ▶ ▶ ▶