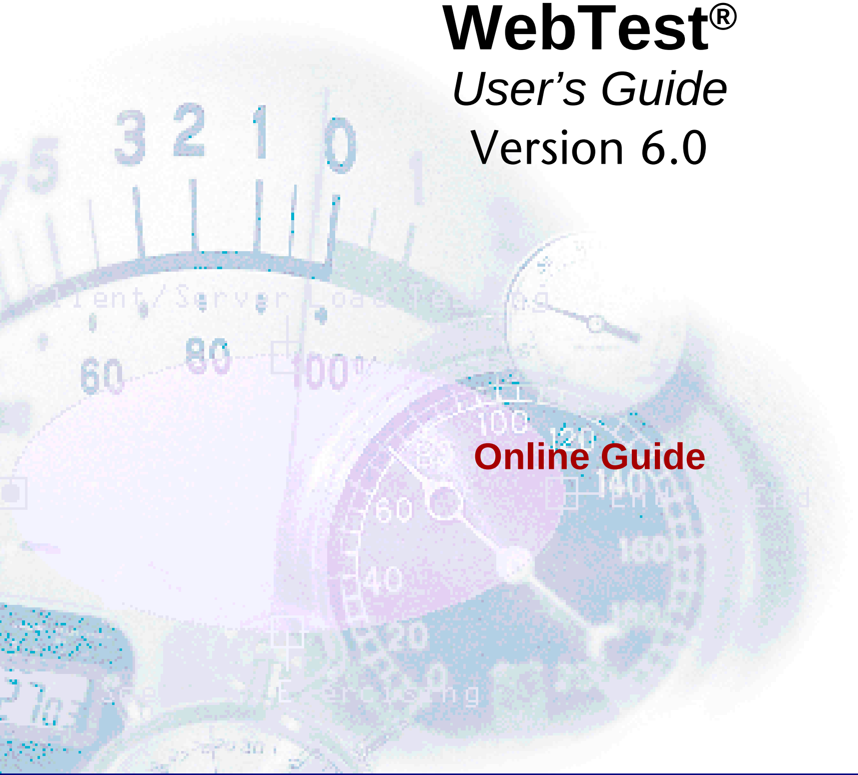


WebTest®

User's Guide

Version 6.0

Online Guide

Books Online

Find

Find Again

Help

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Introduction

Welcome to WebTest, Mercury Interactive's automated software testing tool for Web sites.

This chapter describes:

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Using WebTest

WebTest is an add-in to WinRunner, Mercury Interactive’s automated GUI testing tool for Microsoft Windows applications. WebTest enables you to test and verify the functionality of your Web site while using the Netscape Navigator or the Microsoft Internet Explorer Web browser. You can create a suite of tests and then run the tests each time you change your Web site. To create a test, use WinRunner to record the operations you perform on your Web site. As you click on hypertext and hypergraphic links, WinRunner generates a test script in TSL, Mercury Interactive's C-like test script language.

You can further enhance the recorded test script by inserting checkpoints. A checkpoint verifies the text content, links, images, tables, and standard properties of a Web page.

This guide explains how to use WebTest to test Web sites viewed using the Netscape Navigator or the Microsoft Internet Explorer Web browser. It should be used in conjunction with the *WinRunner User’s Guide* and *TSL Online Reference*.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

How WebTest Identifies Web Objects

WinRunner learns a set of default properties for each object you operate on while recording a test. These properties enable WinRunner to obtain a unique identification for every object that you test. This information is stored in the GUI map. WinRunner uses the GUI map to help it locate frames and objects during a test run.

WinRunner identifies each HTML page and frame that it encounters as a separate window. The *class* property indicates the class of the object. The *html_name* property indicates the name assigned to the HTML page or frame. The *MSW_class* property indicates the MSW_class to which the HTML page or frame belong.

For example, a Web page may have the following information in the GUI map:

```
{
class: window,
MSW_class: html_frame,
html_name: "Mercury Interactive Home Page"
}
```

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

WinRunner also assigns the *html_name* property to standard objects. For example, a search button may have the following information in the GUI map:

```
{
class: push_button,
MSW_class: html_push_button,
html_name: Search
}
```

Web objects such as hypertext links, image links, and images may include the following information in the GUI map.

```
{
class: object,
MSW_class: html_text_link,
html_name: "Mercury Interactive’s Products"
}
```

You can view the contents of your GUI map files in the GUI Map Editor, by choosing **Tools > GUI Map Editor**. The GUI Map Editor displays the logical names and the physical descriptions of objects. For more information on GUI maps, refer to the “Understanding the GUI Map” section in the *WinRunner User’s Guide*.

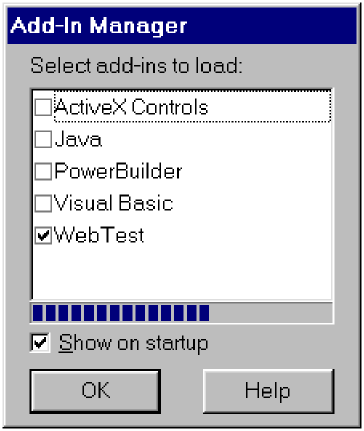
Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Starting WebTest

Before you begin testing your Web site, make sure that you have installed all the necessary files and made any necessary configuration changes. For more information, refer to the *WinRunner Installation Guide*.

To start WebTest:

1 Click **Programs > WinRunner > WinRunner** in the Start menu. The **Add-in Manager** dialog box opens.



Add-In Manager

Select add-ins to load:

☐ ActiveX Controls
☐ Java
☐ PowerBuilder
☐ Visual Basic
☑ WebTest

☑ Show on startup

2 Select **WebTest**.

3 Click **OK**. WinRunner opens with the WebTest add-in loaded.

For more information on the Add-in Manager, refer to the *WinRunner User's Guide.*

Books Online
Find
Find Again
Help
Top of Chapter
Back

You can quickly create a test script by recording the operations you perform in your Web browser.

This chapter describes:

- **Planning Tests**
- **Recording Tests**
- **Understanding Your Test Script**
- **Enhancing WebTest Scripts with TSL**

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

About Creating Tests

You can create a test by recording, programming, or a combination of both methods. The easiest way to create a test is by recording. When you record a test, the operations that you perform on a Web site are recorded in the test script as statements in Test Script Language (TSL). Usually you create a script by recording, and then you use programming to enhance the recorded script.

You can further increase the power of your test scripts by adding checkpoints that verify Web objects such as frames, tables, cells, links, images, and text.

Planning Tests

Before you start recording, you should plan your test. You should consider the following:

- The functionality you want to test. Short tests that check specific functions of the application or complete a transaction are better than long tests that perform several tasks.
- The information you want to validate during the test. A checkpoint can check for differences in the HTML text, links, tables, and standard properties of a page. For more information, see Chapter 3, **Checking Web Pages**.

For more information on planning tests, refer to the “Creating Tests” section in the *WinRunner User’s Guide*. If you are using TestDirector to organize the testing process, you can also refer to the “Planning Tests” section in the *TestDirector User’s Guide.*

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Recording Tests

After planning your test, you are ready to start recording your test script using WinRunner's Context Sensitive recording mode. In this mode, WinRunner records the operations you perform on your Web site and automatically generates a test script.

To create a test script:

1 Start WinRunner.

2 Start your Web browser.

Note: You must start WinRunner before you start your browser. Otherwise, WinRunner will not record and run your test script properly.

---

3 In WinRunner, choose **File > New**, or click the **New** button to create a new test.

4 Choose **Create > Record-Context Sensitive**, or click the **Record** button. WinRunner starts recording your operations.

5 Perform the sequence in your Web browser that you want to record.

As you record, each operation you perform generates a TSL statement in your test script.

Books Online
Find
Find Again
Help
Top of Chapter
Back

You can insert checkpoints in your test by choosing **Create > GUI Checkpoint > For Object/Window or Create > GUI Checkpoint > For Multiple Objects**. For more information, see Chapter 3, **Checking Web Pages**.

6 To stop recording, choose **Create > Stop Recording** or click the **Stop** button.

7 To save your test, choose **File > Save As** and assign the test a name.

Note: As you record, WinRunner also stores information about each object (HTML text, links, images, tables, and frames) in a GUI map file. It is important to save this GUI map file after you record. For more information on GUI map files, refer to the “Creating the GUI Map” section in the *WinRunner User’s Guide.*

If you recorded test scripts with WinRunner before installing the WebTest Add-in, clear the GUI map before creating any WebTest scripts. In WinRunner choose **Tools > GUI Map Editor** to open the GUI Map Editor. Then choose **Edit > Clear All** and close the GUI Map Editor.

Books Online
Find
Find Again
Help
Top of Chapter
Back

Understanding Your Test Script

As you record, each operation you perform generates a statement in Mercury Interactive's Test Script Language (TSL), in your test script. The following is a sample of a recorded WinRunner test script.

```
set_window("Mercury Interactive Home Page", 10);
web_link_click("Products");
set_window("Mercury Interactive Products", 10);
web_image_click("WebTest", 48, 6);
```

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

set_window Function

Each time you click a link in a page, WinRunner generates a **set_window** statement. The **set_window** statement directs input to the currently displayed page in the browser. It has the following syntax:

set_window (*window, time* **);**

For example, the statement:

```
set_window ( "Mercury Interactive Home Page", 10 );
```

indicates that "Mercury Interactive Home Page" is the name of the current Web page. WinRunner waits a maximum of 10 seconds for the page to open, plus the timeout value defined in the General Options dialog box (**Settings > General Options)**.

For more information on the **set_window** function, refer to the *TSL Online Reference* (**Help > TSL Online Reference**).

web_link_click Function

WinRunner records this function when you click a hypertext link. It has the following syntax:

web_link_click (*link_name* **);**

For example, the statement:

```
web_link_click ( "Products" );
```

tells WinRunner to click on the "Products" hypertext link.

For more information on the **web_link_click** function, refer to the *TSL Online Reference* (**Help > TSL Online Reference**).

web_image_click Function

WinRunner records this function when you click a hypergraphic link or an image. It has the following syntax:

web_image_click (*image_name, x, y* **);**

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
⇦ Back

For example, the statement:

```
web_image_click( "WinRunner", 48, 6 );
```

tells WinRunner to click an image called "WinRunner". This image can be a hypergraphic link or a bitmap. 48 and 6 are the coordinates of the mouse pointer when clicked on the image. The coordinates are relative to the upper left corner of the image.

For more information on the **web_image_click** function, refer to the *TSL Online Reference* (**Help > TSL Online Reference**).

Books Online
Find
Find Again
Help
Top of Chapter
Back

Enhancing WebTest Scripts with TSL

You can enhance your recorded test scripts by adding **web_** TSL functions. In the Function Generator, the **web_** functions are located in the *Web* category. For information on the Function Generator, refer to Chapter 20, “Generating Functions” in the *WinRunner User’s Guide*.

The following **web_** TSL functions are available. Note that you can find additional information about these functions in the *TSL Online Reference* (**Help > TSL Online Reference**).

- The **web_browser_invoke** function invokes the browser and opens the specified site. It has the following syntax:

  web_browser_invoke (*browser, site* **);**

- The **web_cursor_to_image** function directs the cursor to move to an image on a page. It has the following syntax:

  web_cursor_to_image (*image_name, x, y* **);**

- The **web_cursor_to_label** function directs the cursor to move to a label on a page. It has the following syntax:

  web_cursor_to_label (*label_name, x, y* **);**

- The **web_cursor_to_link** function directs the cursor to move to a hypertext link on a page. It has the following syntax:

  web_cursor_to_link (*link_name, x, y* **);**

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

- The **web_event** function runs an event on a specified object. It has the following syntax:

  web_event (*object, event_name, x, y* **);**

- The **web_file_browse** function directs the cursor to click on a browse button. It has the following syntax:

  web_file_browse (*web_object* **);**

- The **web_file_set** function sets the text value in a file type object. It has the following syntax:

  web_file_set (*web_object, value* **);**

- The **web_frame_get_text** function retrieves the text content of a page. It has the following syntax:

  web_frame_get_text (*frame_name, text* **);**

- The **web_get_run_event_mode** function returns the current run mode. It has the following syntax:

  web_get_run_event_mode (*out_mode* **);**

- The **web_get_timeout** function returns the maximum time that WinRunner waits for response from the Web. It has the following syntax:

  web_get_timeout (*out_timeout* **);**

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

- The **web_image_click** function is recorded when you click a hypergraphic link or an image. It has the following syntax:

  web_image_click (*image_name, x, y* **);**

- The **web_label_click** function performs a click on a specified label. It has the following syntax:

  web_label_click (*label* **);**

- The **web_link_click** function is recorded when you click a hypertext link. It has the following syntax:

  web_link_click (*link_name* **);**

- The **web_obj_get_child_item** function returns the description of the children in a table object. It has the following syntax:

  web_obj_get_child_item (*web_object, table_row, table_column, object_type, index, object* **);**

- The **web_obj_get_child_item_count** function returns the count of the children in a table object. It has the following syntax:

  web_obj_get_child_item_count (*web_object, table_row, table_column, object_type, object_count* **);**

- The **web_obj_get_info** function returns the value of an object property. It has the following syntax:

  web_obj_get_info (*web_object, property_name, property_value* **);**

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
⇦ Back

- The **web_restore_event_default** function restores the default settings for an event. It has the following syntax:

  web_restore_event_default ( );

- The **web_set_event** function sets the event status. It has the following syntax:

  web_set_event (*class, event_name, event_type, event_status* **);**

- The **web_set_run_event_mode** function sets the event run mode. It has the following syntax:

  web_set_run_event_mode (*mode* **);**

- The **web_set_timeout** function sets the maximum time WinRunner waits for a response from the Web. It has the following syntax:

  web_set_timeout (*timeout* **);**

- The **web_sync** function is recorded when you navigate between links within the same frame. It has the following syntax:

  web_sync (*timeout* **);**

- The **web_text_exists** function returns text value if it is found in the frame. It has the following syntax:

  web_text_exists (*frame, text_to_find, parent_object, table_row, table_column* **);**

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

When testing tables, you can also use the following **tbl_get** functions:

- The **tbl_get_cell_data** function retrieves the contents of the specified cell from a table. It has the following syntax:

  tbl_get_cell_data (*table, row, column, out_text* **);**

- The **tbl_get_cols_count** function retrieves the number of columns in a table. It has the following syntax:

  tbl_get_cols_count (*table, out_cols_count* **);**

- The **tbl_get_column_name** function retrieves the column header name of the specified column in a table. It has the following syntax:

  tbl_get_column_name (*table, col_index, out_col_names* **);**

- The **tbl_get_rows_count** function retrieves the number of rows in the specified table. It has the following syntax:

  tbl_get_rows_count (*table, out_rows_count* **);**

In the Function Generator, the **tbl_get_** functions are located in the *Table Functions* category.

For more information on TSL functions, refer to the *TSL Online Reference*.

Books Online
Find
Find Again
Help
Top of Chapter
Back

Checking Web Pages

By adding checkpoints to your test scripts, you can compare the behavior of Web objects in different versions of your Web page.

This chapter describes:

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

About Checking Web Pages

Use checkpoints in your test script to help you examine your Web site and detect defects. Using the Check GUI dialog box, you can check frames, tables, cells, links, and images on a Web page for differences between test runs.

You can define checkpoints according to default properties recommended by WinRunner, or you can define custom checks by selecting other properties.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
⇦ Back

Checking Standard Frame Properties

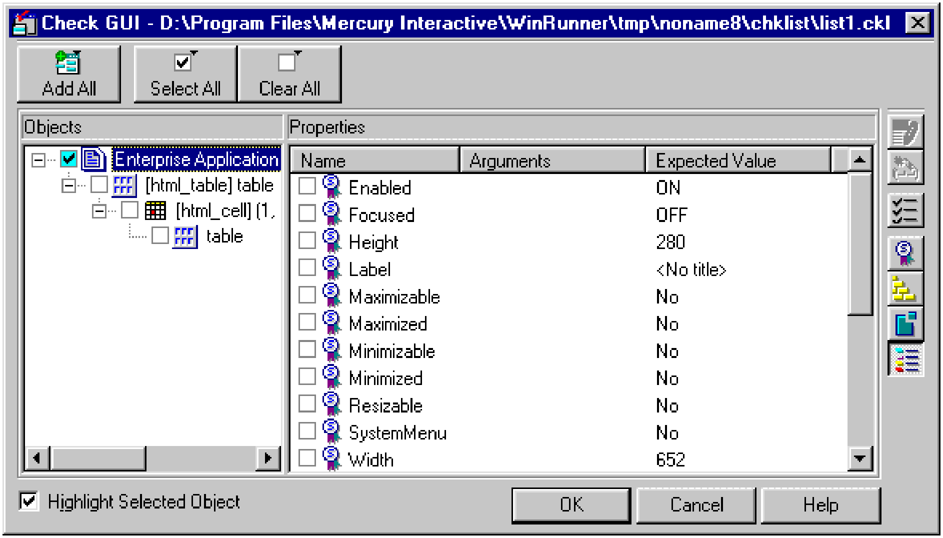
You can create a checkpoint to check standard properties of a frame.

To check standard frame properties:

1 Choose **Create > GUI Checkpoint > For Object/Window.**

The WinRunner window is minimized to an icon, the mouse pointer turns into a pointing hand, and a help window opens.

2 Double-click an object on your Web page. The **Check GUI** dialog box opens, and the object is highlighted.



Check GUI - D:\Program Files\Mercury Interactive\WinRunner\tmp\noname8\chklist\list1.ckl

Name	Arguments	Expected Value
□ Enabled		ON
□ Focused		OFF
□ Height		280
□ Label		<No title>
□ Maximizable		No
□ Maximized		No
□ Minimizable		No
□ Minimized		No
□ Resizable		No
□ SystemMenu		No
□ Width		652

☑ Highlight Selected Object

Books Online
Find
Find Again
Help
Top of Chapter
Back

3 In the **Objects** column, make sure that the frame is selected.

The Properties column indicates the available standard properties and the default check for that frame.

4 In the **Properties** column, choose the properties you want WinRunner to check.

You can check the following standard properties:

- **Enabled** checks whether the frame can be selected.
- **Focused** checks whether keyboard input will be directed to this frame.
- **Label** checks the frame's label.
- **Minimizable** and **Maximizable** check whether the frame can be minimized or maximized.
- **Minimized** and **Maximized** check whether the frame is minimized or maximized.
- **Resizable** checks whether the frame can be resized.
- **SystemMenu** checks whether the frame has a system menu.
- **Width** and **Height** check the frame's width and height, in pixels.
- **X** and **Y** check the x and y coordinates of the top left corner of the frame.

5 Click **OK** to close the dialog box.

Books Online
Find
Find Again
Help
Top of Chapter
Back

WinRunner captures the object information and stores it in the test's expected results folder. The WinRunner window is restored and a checkpoint appears in your test script as a **win_check_gui** statement. For more information on **win_check_gui**, refer to the *TSL Online Reference* (**Help > TSL Online Reference**).

Books Online
Find
Find Again
Help
Top of Chapter
Back

Checking Broken Links

You can create a checkpoint to check whether a text link or an image link is active. You can create a checkpoint to check a single broken link or all the broken links in a frame.

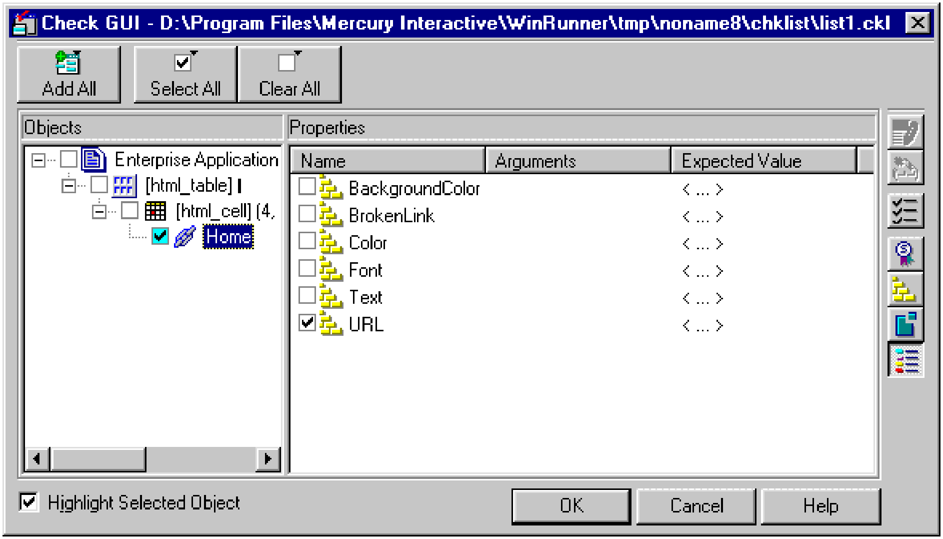
To check a single broken link:

1 Choose **Create > GUI Checkpoint > For Object/Window**.

The WinRunner window is minimized to an icon, the mouse pointer turns into a pointing hand, and a help window opens.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

2 Double-click a link on your Web page. The **Check GUI** dialog box opens, and the object is highlighted.



Check GUI - D:\Program Files\Mercury Interactive\WinRunner\tmp\noname8\chklist\list1.ckl

- □ Enterprise Application
  - □ [html_table] I
    - □ [html_cell] (4,
      - ☑ Home

Name	Arguments	Expected Value
□ BackgroundColor		< ... >
□ BrokenLink		< ... >
□ Color		< ... >
□ Font		< ... >
□ Text		< ... >
☑ URL		< ... >

☑ Highlight Selected Object

3 In the **Objects** column, make sure that the link is selected.

The Properties column indicates the properties available for you to check.

4 In the **Properties** column, select the **BrokenLink** check box.

5 To edit the expected value of the property, click **BrokenLink.** The property is highlighted.

Books Online
Find
Find Again
Help
Top of Chapter
Back

6. Click the **Edit Expected Value** button, or double-click the value in the **Expected Value** column to edit it. A combo box opens.
7. Select **Valid** or **NotValid**. Valid indicates that the link is active, and NotValid indicates that the link is broken.
8. Click **OK** to close the Check GUI dialog box.

   WinRunner captures the object information and stores it in the test's expected results folder. The WinRunner window is restored and a checkpoint appears in your test script as an **obj_check_gui** or **win_check_gui** statement. For more information on **obj_check_gui** and **win_check_gui** functions, refer to the *TSL Online Reference* (**Help > TSL Online Reference**).

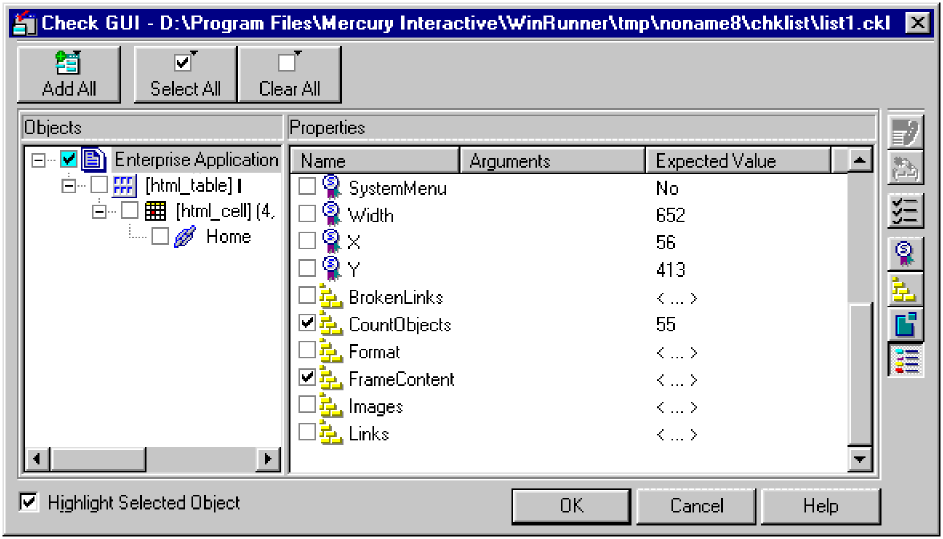
To check all broken links in a frame:

1. Choose **Create > GUI Checkpoint > For Object/Window**.

   The WinRunner window is minimized to an icon, the mouse pointer turns into a pointing hand, and a help window opens.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

2 Double-click an object on your Web page. The **Check GUI** dialog box opens, and an object is highlighted.



Check GUI - D:\Program Files\Mercury Interactive\WinRunner\tmp\noname8\chklist\list1.ckl
Add All
Select All
Clear All
Objects
Properties
OK
Cancel
Help

Objects:

- ☑ Enterprise Application
  - □ [html_table] I
    - □ [html_cell] (4,
      - □ Home

Name	Arguments	Expected Value
□ SystemMenu		No
□ Width		652
□ X		56
□ Y		413
□ BrokenLinks		< ... >
☑ CountObjects		55
□ Format		< ... >
☑ FrameContent		< ... >
□ Images		< ... >
□ Links		< ... >

☑ Highlight Selected Object

3 In the **Objects** column, make sure that frame is selected.

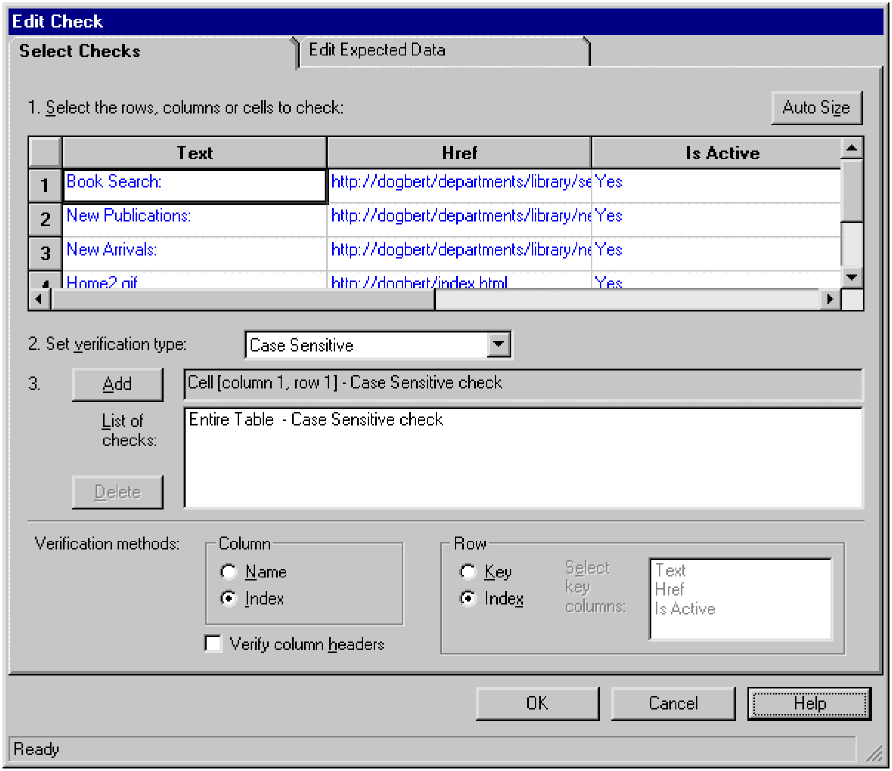
The Properties column indicates the properties available for you to check.

4 In the **Properties** column, select the **BrokenLinks** check box.

5 To edit the expected value of a the property, click **BrokenLinks**. The property is highlighted.

Books Online
Find
Find Again
Help
Top of Chapter
Back

6 Click the **Edit Expected Value** button, or double-click the value in the **Expected Value** column to edit it. The **Edit Check** dialog box opens.



1. Select the rows, columns or cells to check:

	Text	Href	Is Active
1	Book Search:	http://dogbert/departments/library/se	Yes
2	New Publications:	http://dogbert/departments/library/ne	Yes
3	New Arrivals:	http://dogbert/departments/library/ne	Yes
4	Home2.gif	http://dogbert/index.html	Yes

2. Set verification type: Case Sensitive

3. Add — Cell [column 1, row 1] - Case Sensitive check

List of checks: Entire Table - Case Sensitive check

Delete

Verification methods:

Column: ○ Name ● Index

☐ Verify column headers

Row: ○ Key ● Index

Select key columns: Text, Href, Is Active

Ready

Books Online
Find
Find Again
Help
Top of Chapter
Back

The Select Checks view lists the links according to the order they appear in your source code. You can specify which links to check, and which verification method and verification type to use. The Select Checks view includes the following columns:

- **Text** indicates the name of the link.
- **Href** indicates the address of the link.
- **Is Active** indicates whether the link is active.

7 The default check for a multiple-column table is a case sensitive check on the entire table by column name and row index. The default check for a single-column table is a case sensitive check on the entire table by row position. If you do not wish to accept the default settings, you must delete the default check before you specify the checks to perform. Select the **Entire Table - Case Sensitive check** entry in the **List of Checks** box and click the **Delete** button. Alternatively, double-click this entry in the **List of Checks** box. A WinRunner message prompts you to delete the highlighted check. Click **Yes**.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

8 Select the type of check from the **Verification Type** list.

WinRunner can verify the links in several different ways. You can choose different verification types for different selections of cells.

- **Case Sensitive** (the default): WinRunner compares the text content of the selection. Any difference in case or text content between the expected and actual data results in a mismatch.
- **Case Insensitive:** WinRunner compares the text content of the selection. Only differences in text content between the expected and actual data result in a mismatch.
- **Numeric Content:** WinRunner evaluates the selected data according to numeric values. WinRunner recognizes, for example, that “2” and “2.00” are the same number.
- **Numeric Range:** WinRunner compares the selected data against a numeric range. Both the minimum and maximum values are any real number that you specify. This comparison differs from text and numeric content verification in that the actual table data is compared against the range that you defined and not against the expected results.
- **Case Sensitive Ignore Spaces:** WinRunner checks the data in the cell according to case and content, ignoring differences in spaces. WinRunner reports any differences in case or content as a mismatch.
- **Case Insensitive Ignore Spaces:** WinRunner checks the content in the cell according to content, ignoring differences in case and spaces. WinRunner reports only differences in content as a mismatch.

Books Online
Find
Find Again
Help
Top of Chapter
Back

9 Specify the parts of the table that you want to check.

Highlight the cells on which you want to perform the content check. Next, click the **Add** button toolbar to add a check for these cells. Alternatively, you can:

- double-click a cell to check it
- double-click a row header to check all the cells in a row
- double-click a column header to check all the cells in a column
- double-click the top-left corner to check the entire table

A description of the cells to be checked appears in the **List of Checks** box.

10 Specify the verification method.

You can select the verification method for controlling how WinRunner identifies columns or rows within a table. The verification method applies to the entire table. Specifying the verification method is different for multiple-column and single-column tables.

Books Online
Find
Find Again
Help
Top of Chapter
Back

The following verification methods are available for a multiple-column table:

Column

- **Name:** WinRunner looks for the selection according to the column names. A shift in the position of the columns within the table does not result in a mismatch.
- **Index:** WinRunner looks for the selection according to the index, or position, of the columns. A shift in the position of the columns within the table results in a mismatch. Select this option if your table contains multiple columns with the same name. Choosing this option enables the **Verify column headers** check box, which enables you to check column headers as well as cells.

Row

- **Key:** WinRunner looks for the rows in the selection according to the key(s) specified in the **Select key columns** list box, which lists the names of all columns in the table. A shift in the position of any of the rows does not result in a mismatch. If the key selection does not identify a unique row, only the first matching row will be checked.
- **Index** (default setting): WinRunner looks for the selection according to the index, or position, of the rows. A shift in the position of any of the rows results in a mismatch.

Books Online
Find
Find Again
Help
Top of Chapter
Back

The following verification methods are available for a single-column table:

- **By Position:** WinRunner checks the selection according to the location of the items within the column.
- **By Content:** WinRunner checks the selection according to the content of the items, ignoring their location in the column.

11 Click **OK** to save and close the Edit Check dialog box. The Check GUI dialog box is restored.

WinRunner captures the object information and stores it in the test's expected results folder. The WinRunner window is restored and a checkpoint appears in your test script as **win_check_gui** statement. For more information on **win_check_gui** functions, refer to the *TSL Online Reference* (**Help > TSL Online Reference**).

Books Online
Find
Find Again
Help
Top of Chapter
Back

Checking the Object Count in Frames

You can create a checkpoint to check the number of objects in a frame.

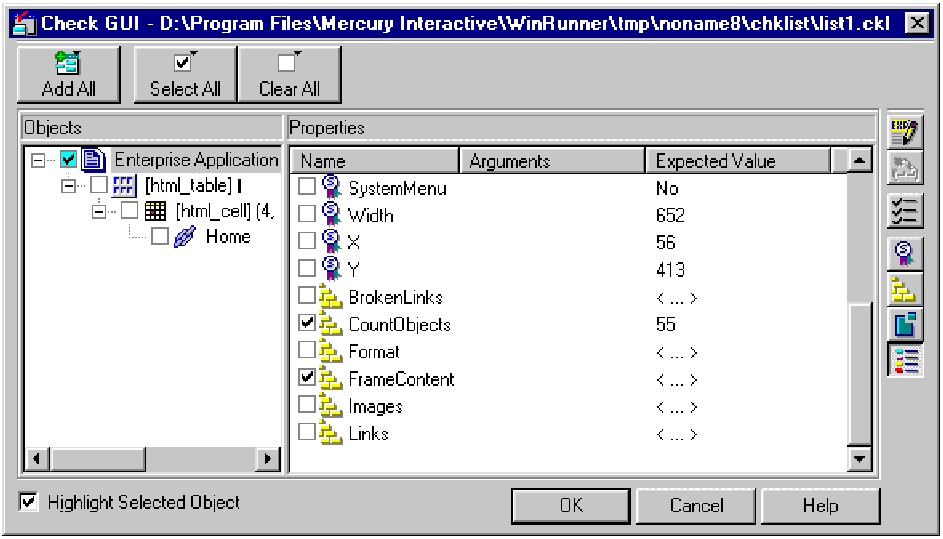
To check the object count in a frame:

1 Choose **Create > GUI Checkpoint > For Object/Window**.

The WinRunner window is minimized to an icon, the mouse pointer turns into a pointing hand, and a help window opens.

Books Online
Find
Find Again
Help
Top of Chapter
Back

2 Double-click an object on your Web page. The **Check GUI** dialog box opens, and the object is highlighted.



Check GUI - D:\Program Files\Mercury Interactive\WinRunner\tmp\noname8\chklist\list1.ckl

- ☑ Enterprise Application
  - □ [html_table] I
    - □ [html_cell] (4,
      - □ Home

	Name	Arguments	Expected Value
□	SystemMenu		No
□	Width		652
□	X		56
□	Y		413
□	BrokenLinks		< ... >
☑	CountObjects		55
□	Format		< ... >
☑	FrameContent		< ... >
□	Images		< ... >
□	Links		< ... >

☑ Highlight Selected Object

3 In the **Objects** column, make sure that the frame is selected.

The Properties column indicates the properties available for you to check.

4 In the **Properties** column, select the **CountObjects** check box.

5 To edit the expected value of the property, click **CountObjects**. The property is highlighted.

Books Online
Find
Find Again
Help
Top of Chapter
Back

6. Click the **Edit Expected Value** button, or double-click the value in the **Expected Value** column to edit it. A spin box opens.

7. Enter the expected number of objects.

8. Click **OK** to close the Check GUI dialog box.

   WinRunner captures the object information and stores it in the test's expected results folder. The WinRunner window is restored and a checkpoint appears in your test script as **win_check_gui** statement. For more information on **win_check_gui** functions, refer to the *TSL Online Reference* (**Help > TSL Online Reference**).

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Checking the Structure of Frames, Tables, and Cells

You can create a checkpoint to check the structure of frames, tables, and cells on a Web page.

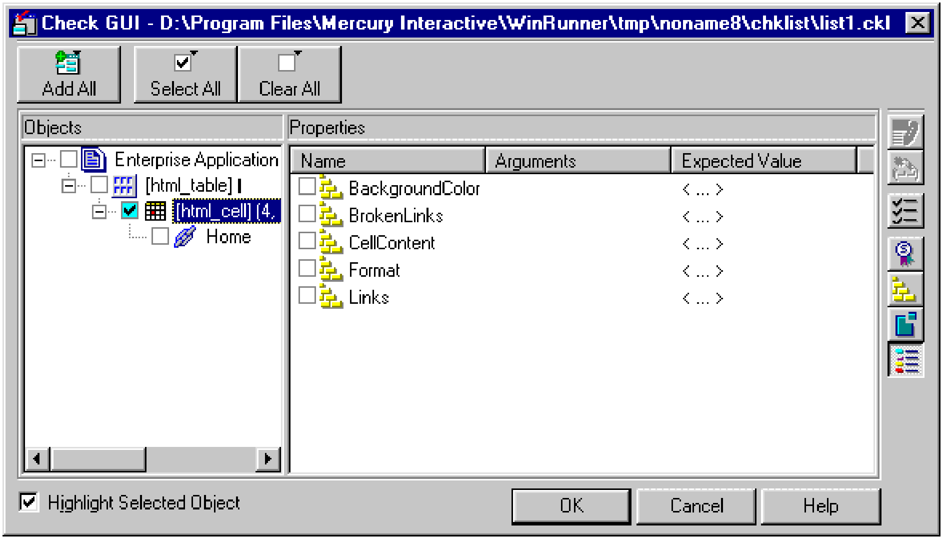
To check the structure of a frame, table, or cell:

1 Choose **Create > GUI Checkpoint > For Object/Window**.

The WinRunner window is minimized to an icon, the mouse pointer turns into a pointing hand, and a help window opens.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

2 Double-click an object on your Web page. The **Check GUI** dialog box opens, and the object is highlighted.



Check GUI - D:\Program Files\Mercury Interactive\WinRunner\tmp\noname8\chklist\list1.ckl

- □ Enterprise Application
  - □ [html_table] I
    - ☑ [html_cell] (4,
      - □ Home

Name	Arguments	Expected Value
□ BackgroundColor		< ... >
□ BrokenLinks		< ... >
□ CellContent		< ... >
□ Format		< ... >
□ Links		< ... >

☑ Highlight Selected Object

3 In the **Objects** column, select an object.

The Properties column indicates the properties available for you to check.

4 In the **Properties** column, select the **Format** check box.

5 To edit the expected value of the property, click **Format.** The property is highlighted.

Books Online
Find
Find Again
Help
Top of Chapter
Back

6. Click the **Edit Expected Value** button, or double-click the value in the **Expected Value** column to edit it. A text file opens in Notepad describing the structure of the frame, table, or cell.
7. Modify the expected structure.
8. Save the text file and close Notepad.
9. Click **OK** to close the Check GUI dialog box.

   WinRunner captures the object information and stores it in the test's expected results folder. The WinRunner window is restored and a checkpoint appears in your test script as an **obj_check_gui** or **win_check_gui** statement. For more information on **obj_check_gui** and **win_check_gui** functions, refer to the *TSL Online Reference* (**Help > TSL Online Reference**).

Books Online
Find
Find Again
Help
Top of Chapter
Back

Checking the Content of Frames, Cells, Links, or Images

You can create a checkpoint to check the content of a frame, cell, text link, image link, or an image. To check the content of a table, see **Checking the Content of Tables** on page 47.

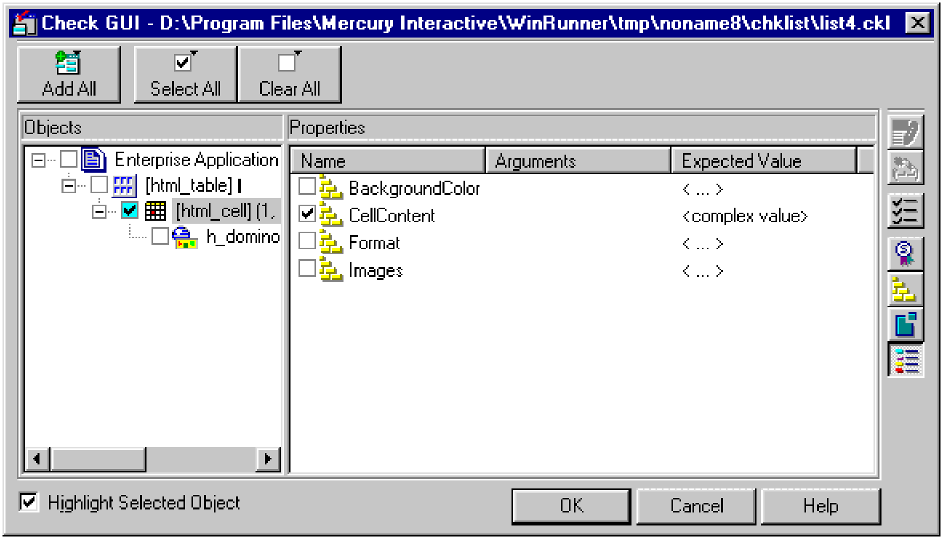
To check content:

1 Choose **Create > GUI Checkpoint > For Object/Window**.

The WinRunner window is minimized to an icon, the mouse pointer turns into a pointing hand, and a help window opens.

Books Online
Find
Find Again
Help
Top of Chapter
Back

2 Double-click an object on your Web page. The **Check GUI** dialog box opens, and the object is highlighted.


Check GUI - D:\Program Files\Mercury Interactive\WinRunner\tmp\noname8\chklist\list4.ckl
Add All
Select All
Clear All
Objects
Properties
☑ Highlight Selected Object
OK
Cancel
Help

- □ Enterprise Application
  - □ [html_table] I
    - ☑ [html_cell] (1,
      - □ h_domino

Name	Arguments	Expected Value
□ BackgroundColor		< ... >
☑ CellContent		<complex value>
□ Format		< ... >
□ Images		< ... >

3 In the **Objects** column, select an object (frame, cell, text link, image link, or an image).

The Properties column indicates the properties available for you to check.

Books Online
Find
Find Again
Help
Top of Chapter
Back

4 In the **Properties** column, select one of the following checks:

- If your object is a frame, select the **FrameContent** check box.
- If your object is a cell, select the **CellContent** check box.
- If your object is a text link, select the **Text** check box.
- If your object is an image link, select the **ImageContent** check box.
- If your object is an image, select the **ImageContent** check box.

5 To edit the expected value of a the property, click a property. The property is highlighted.

Note that you cannot edit the expected value of the **ImageContent** property.

6 Click the **Edit Expected Value** button, or double-click the value in the **Expected Value** column to edit it.

- For the **FrameContent** property, an editor opens.
- For the **CellContent** property, an editor opens.
- For the **Text** property, an edit box opens.

7 Modify the expected value.

Books Online
Find
Find Again
Help
Top of Chapter
Back

8 Click **OK** to close the Check GUI dialog box.

WinRunner captures the object information and stores it in the test's expected results folder. The WinRunner window is restored and a checkpoint appears in your test script as an **obj_check_gui** or **win_check_gui** statement. For more information on **obj_check_gui** and **win_check_gui** functions, refer to the *TSL Online Reference* (**Help > TSL Online Reference**).

Checking the Content of Tables

You can create a checkpoint to check the text content of a table.

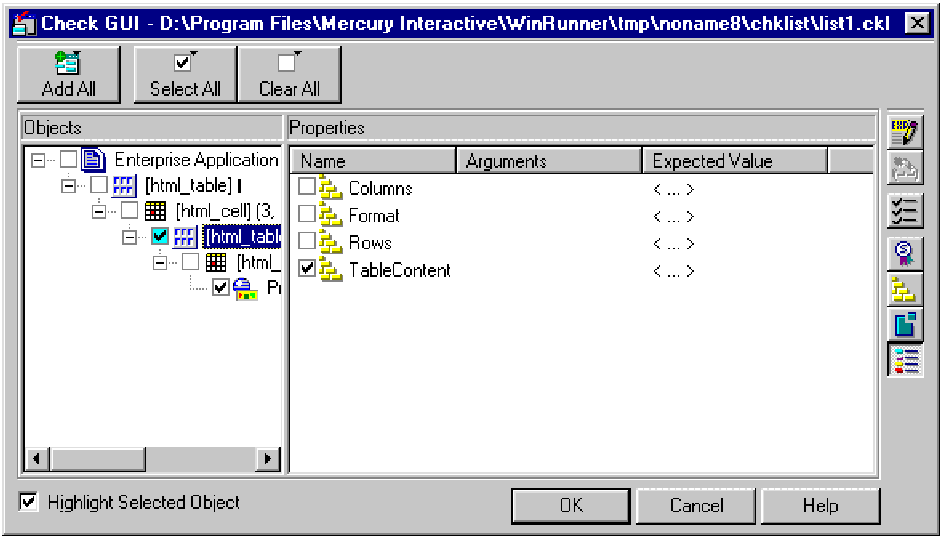
To check the content of a table:

1 Choose **Create > GUI Checkpoint > For Object/Window**.

The WinRunner window is minimized to an icon, the mouse pointer turns into a pointing hand, and a help window opens.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

2 Double-click a table on your Web page. The **Check GUI** dialog box opens, and an object is highlighted.



Check GUI - D:\Program Files\Mercury Interactive\WinRunner\tmp\noname8\chklist\list1.ckl

Name	Arguments	Expected Value
□ Columns		< ... >
□ Format		< ... >
□ Rows		< ... >
☑ TableContent		< ... >

☑ Highlight Selected Object

3 In the **Objects** column, make sure that the table is selected.

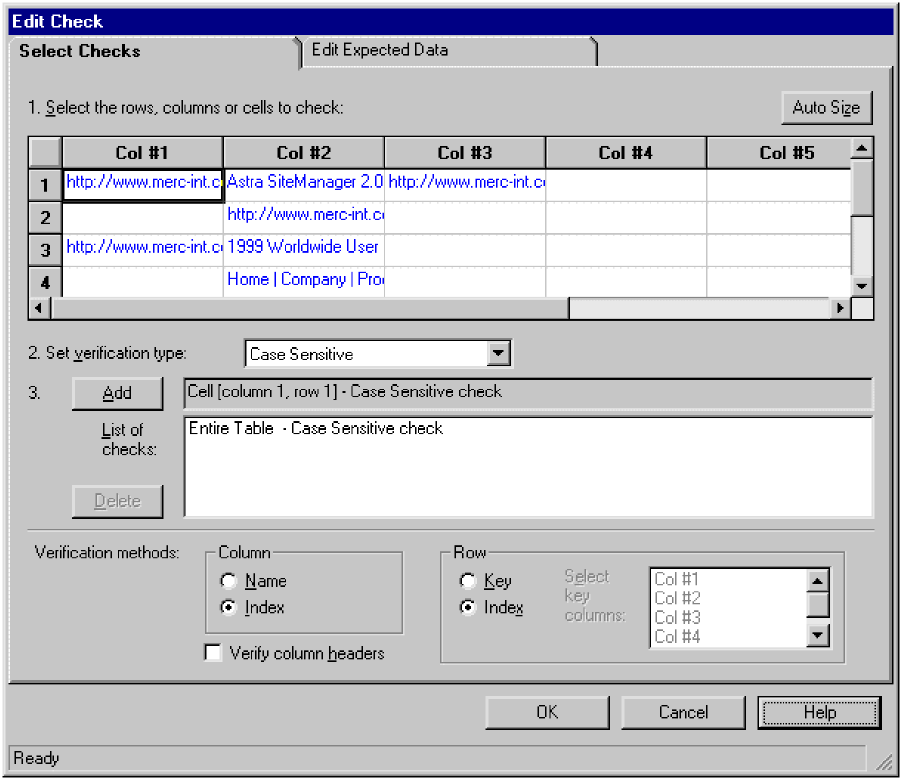
The Properties column indicates the properties available for you to check.

4 In the **Properties** column, select the **TableContent** check box.

5 To edit the expected value of a the property, click **TableContent**. The property is highlighted.

Books Online
Find
Find Again
Help
Top of Chapter
Back

6 Click the **Edit Expected Value** button, or double-click the value in the **Expected Value** column to edit it. The **Edit Check** dialog box opens.



Edit Check

1. Select the rows, columns or cells to check:

	Col #1	Col #2	Col #3	Col #4	Col #5
1	http://www.merc-int.c	Astra SiteManager 2.0	http://www.merc-int.c		
2		http://www.merc-int.c			
3	http://www.merc-int.c	1999 Worldwide User			
4		Home | Company | Pro			

2. Set verification type: Case Sensitive

3. Add — Cell [column 1, row 1] - Case Sensitive check

List of checks: Entire Table - Case Sensitive check

Verification methods:

Column
○ Name
● Index

□ Verify column headers

Row
○ Key
● Index

Select key columns: Col #1, Col #2, Col #3, Col #4

Ready

Books Online
Find
Find Again
Help
Top of Chapter
Back

The Select Checks view displays the text content of a table. You can specify which column or rows to check in the table, and which verification method and verification type to use.

The default check for a multiple-column table is a case sensitive check on the entire table by column name and row index. The default check for a single-column table is a case sensitive check on the entire table by row position.

7 If you do not wish to accept the default settings, you must delete the default check before you specify the checks to perform. Select the **Entire Table - Case Sensitive check** entry in the **List of Checks** box and click the **Delete** button. Alternatively, double-click this entry in the **List of Checks** box. A WinRunner message prompts you to delete the highlighted check. Click **Yes**.

Books Online
Find
Find Again
Help
Top of Chapter
Back

8 Select the type of check from the **Verification Type** list.

WinRunner can verify the links in several different ways. You can choose different verification types for different selections of cells.

- **Case Sensitive** (the default): WinRunner compares the text content of the selection. Any difference in case or text content between the expected and actual data results in a mismatch.
- **Case Insensitive:** WinRunner compares the text content of the selection. Only differences in text content between the expected and actual data result in a mismatch.
- **Numeric Content:** WinRunner evaluates the selected data according to numeric values. WinRunner recognizes, for example, that “2” and “2.00” are the same number.
- **Numeric Range:** WinRunner compares the selected data against a numeric range. Both the minimum and maximum values are any real number that you specify. This comparison differs from text and numeric content verification in that the actual table data is compared against the range that you defined and not against the expected results.
- **Case Sensitive Ignore Spaces:** WinRunner checks the data in the cell according to case and content, ignoring differences in spaces. WinRunner reports any differences in case or content as a mismatch.
- **Case Insensitive Ignore Spaces:** WinRunner checks the content in the cell according to content, ignoring differences in case and spaces. WinRunner reports only differences in content as a mismatch.

Books Online
Find
Find Again
Help
Top of Chapter
Back

9 Specify the parts of the table that you want to check.

Highlight the cells on which you want to perform the content check. Next, click the **Add** button toolbar to add a check for these cells. Alternatively, you can:

- double-click a cell to check it
- double-click a row header to check all the cells in a row
- double-click a column header to check all the cells in a column
- double-click the top-left corner to check the entire table

A description of the cells to be checked appears in the **List of Checks** box.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

10 Specify the verification method.

You can select the verification method for controlling how WinRunner identifies columns or rows within a table. The verification method applies to the entire table. Specifying the verification method is different for multiple-column and single-column tables.

The following verification methods are available for a multiple-column table:

Column

- **Name:** WinRunner looks for the selection according to the column names. A shift in the position of the columns within the table does not result in a mismatch.
- **Index:** WinRunner looks for the selection according to the index, or position, of the columns. A shift in the position of the columns within the table results in a mismatch. Select this option if your table contains multiple columns with the same name. Choosing this option enables the **Verify column headers** check box, which enables you to check column headers as well as cells.

Books Online
Find
Find Again
Help
Top of Chapter
Back

Row

- **Key:** WinRunner looks for the rows in the selection according to the key(s) specified in the **Select key columns** list box, which lists the names of all columns in the table. A shift in the position of any of the rows does not result in a mismatch. If the key selection does not identify a unique row, only the first matching row will be checked.
- **Index** (default setting): WinRunner looks for the selection according to the index, or position, of the rows. A shift in the position of any of the rows results in a mismatch.

The following verification methods are available for a single-column table:

- **By Position:** WinRunner checks the selection according to the location of the items within the column.
- **By Content:** WinRunner checks the selection according to the content of the items, ignoring their location in the column.

Books Online
Find
Find Again
Help
Top of Chapter
Back

11 Click **OK** to save and close the Edit Check dialog box. The Check GUI dialog box is restored.

WinRunner captures the object information and stores it in the test's expected results directory. The WinRunner window is restored and a checkpoint appears in your test script as **win_check_gui** statement. For more information on **win_check_gui** functions, refer to the *TSL Online Reference* (**Help > TSL Online Reference**).

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Checking Links and Images in a Frame

You can create a checkpoint to check image links, text links and images in a frame.

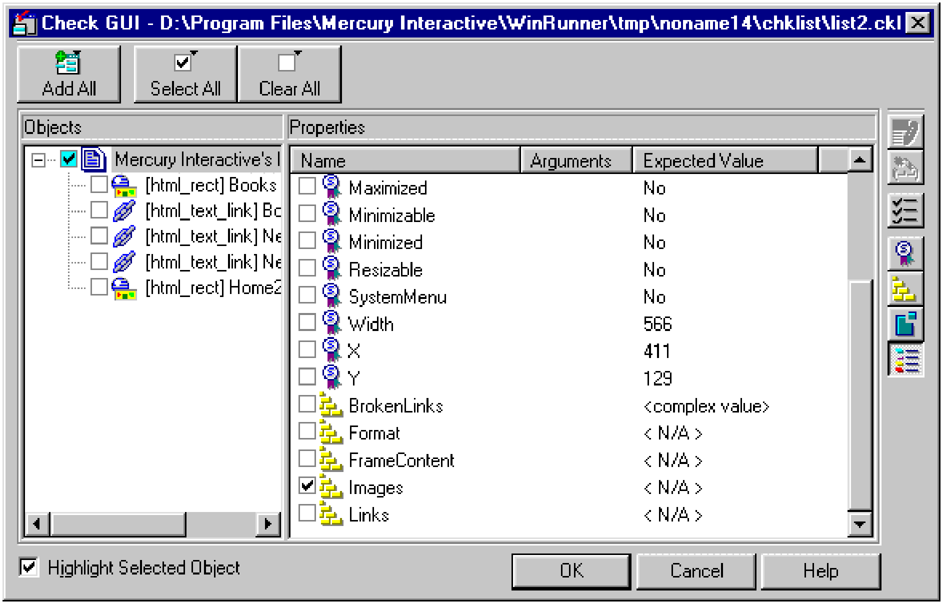
To check links and images in a frame:

1 Choose **Create > GUI Checkpoint > For Object/Window**.

The WinRunner window is minimized to an icon, the mouse pointer turns into a pointing hand, and a help window opens.

Books Online
Find
Find Again
Help
Top of Chapter
Back

2 Double-click an object on your Web page. The **Check GUI** dialog box opens, and an object is highlighted.


Check GUI - D:\Program Files\Mercury Interactive\WinRunner\tmp\noname14\chklist\list2.ckl
Add All
Select All
Clear All
Objects
Properties
OK
Cancel
Help

Objects:

- ☑ Mercury Interactive's I
  - □ [html_rect] Books
  - □ [html_text_link] Bc
  - □ [html_text_link] Ne
  - □ [html_text_link] Ne
  - □ [html_rect] Home2

	Name	Arguments	Expected Value
□	Maximized		No
□	Minimizable		No
□	Minimized		No
□	Resizable		No
□	SystemMenu		No
□	Width		566
□	X		411
□	Y		129
□	BrokenLinks		<complex value>
□	Format		< N/A >
□	FrameContent		< N/A >
☑	Images		< N/A >
□	Links		< N/A >

☑ Highlight Selected Object

3 In the **Objects** column, make sure that frame object is selected.

The Properties column indicates the properties available for you to check.

Books Online
Find
Find Again
Help
Top of Chapter
Back

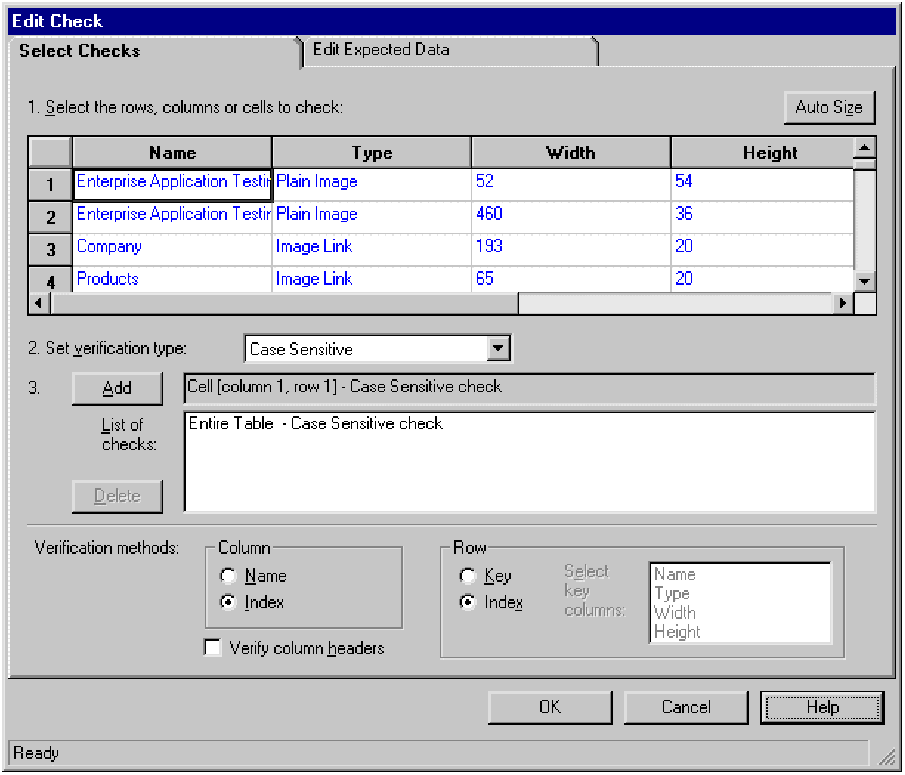
4 In the **Properties** column, select one of the following checks:

- To check images or image links, select the **Images** check box.
- To check text links, select the **Links** check box.

5 To edit the expected value of a the property, click **Images**. The property is highlighted.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

6 Click the **Edit Expected Value** button, or double-click the value in the **Expected Value** column to edit it. The **Edit Check** dialog box opens.



Edit Check

1. Select the rows, columns or cells to check:

	Name	Type	Width	Height
1	Enterprise Application Testir	Plain Image	52	54
2	Enterprise Application Testir	Plain Image	460	36
3	Company	Image Link	193	20
4	Products	Image Link	65	20

2. Set verification type: Case Sensitive

3. Cell [column 1, row 1] - Case Sensitive check

List of checks: Entire Table - Case Sensitive check

Verification methods:

Column: ○ Name ● Index

☐ Verify column headers

Row: ○ Key ● Index

Select key columns: Name, Type, Width, Height

Ready

Books Online
Find
Find Again
Help
Top of Chapter
Back

If you selected the Images property, the Select Checks view lists all images and image links according to the order they appear in your source code. You can specify which images to check, and which verification method and verification type to use. The Select Checks view includes the following columns:

- **Name** indicates the name of the image.
- **Type** indicates whether the image is a plain image, image link, or an image map.
- **Width** and **Height** indicate the dimensions of an image, (only when the image size is defined in the source page).

For the Links property, the Select Checks view list all the text links according to the order they appear in your source code. You can specify which text links to check, and which verification method and verification type to use. The Select Checks view includes the following columns:

- **Text** indicates the name of the link.
- **Href** indicates the address of the link.

7 The default check for a multiple-column table is a case sensitive check on the entire table by column name and row index. The default check for a single-column table is a case sensitive check on the entire table by row position. If you do not wish to accept the default settings, you must delete the default check before you specify the checks to perform. Select the **Entire Table - Case Sensitive check** entry in the **List of Checks** box and click the **Delete** button. Alternatively, double-click this entry in the **List of Checks** box. A WinRunner message prompts you to delete the highlighted check. Click **Yes**.

Books Online
Find
Find Again
Help
Top of Chapter
Back

8 Select the type of check from the **Verification Type** list.

WinRunner can verify the links in several different ways. You can choose different verification types for different selections of cells.

- **Case Sensitive** (the default): WinRunner compares the text content of the selection. Any difference in case or text content between the expected and actual data results in a mismatch.
- **Case Insensitive:** WinRunner compares the text content of the selection. Only differences in text content between the expected and actual data result in a mismatch.
- **Numeric Content:** WinRunner evaluates the selected data according to numeric values. WinRunner recognizes, for example, that “2” and “2.00” are the same number.
- **Numeric Range:** WinRunner compares the selected data against a numeric range. Both the minimum and maximum values are any real number that you specify. This comparison differs from text and numeric content verification in that the actual table data is compared against the range that you defined and not against the expected results.
- **Case Sensitive Ignore Spaces:** WinRunner checks the data in the cell according to case and content, ignoring differences in spaces. WinRunner reports any differences in case or content as a mismatch.
- **Case Insensitive Ignore Spaces:** WinRunner checks the content in the cell according to content, ignoring differences in case and spaces. WinRunner reports only differences in content as a mismatch.

Books Online
Find
Find Again
Help
Top of Chapter
Back

9 Specify the parts of the table that you want to check.

Highlight the cells on which you want to perform the content check. Next, click the **Add** button toolbar to add a check for these cells. Alternatively, you can:

- double-click a cell to check it
- double-click a row header to check all the cells in a row
- double-click a column header to check all the cells in a column
- double-click the top-left corner to check the entire table

A description of the cells to be checked appears in the **List of Checks** box.

10 Specify the verification method.

You can select the verification method for controlling how WinRunner identifies columns or rows within a table. The verification method applies to the entire table. Specifying the verification method is different for multiple-column and single-column tables.

Books Online
Find
Find Again
Help
Top of Chapter
Back

The following verification methods are available for a multiple-column table:

Column

- **Name:** WinRunner looks for the selection according to the column names. A shift in the position of the columns within the table does not result in a mismatch.
- **Index:** WinRunner looks for the selection according to the index, or position, of the columns. A shift in the position of the columns within the table results in a mismatch. Select this option if your table contains multiple columns with the same name. Choosing this option enables the **Verify column headers** check box, which enables you to check column headers as well as cells.

Row

- **Key:** WinRunner looks for the rows in the selection according to the key(s) specified in the **Select key columns** list box, which lists the names of all columns in the table. A shift in the position of any of the rows does not result in a mismatch. If the key selection does not identify a unique row, only the first matching row will be checked.
- **Index** (default setting): WinRunner looks for the selection according to the index, or position, of the rows. A shift in the position of any of the rows results in a mismatch.

The following verification methods are available for a single-column table:

- **By Position:** WinRunner checks the selection according to the location of the items within the column.
- **By Content:** WinRunner checks the selection according to the content of the items, ignoring their location in the column.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

11 Click **OK** to save and close the Edit Check dialog box. The Check GUI dialog box is restored.

WinRunner captures the object information and stores it in the test's expected results directory. The WinRunner window is restored and a checkpoint appears in your test script as **win_check_gui** statement. For more information on **win_check_gui** functions, refer to the *TSL Online Reference* (**Help > TSL Online Reference**).

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Checking the Number of Columns and Rows in a Table

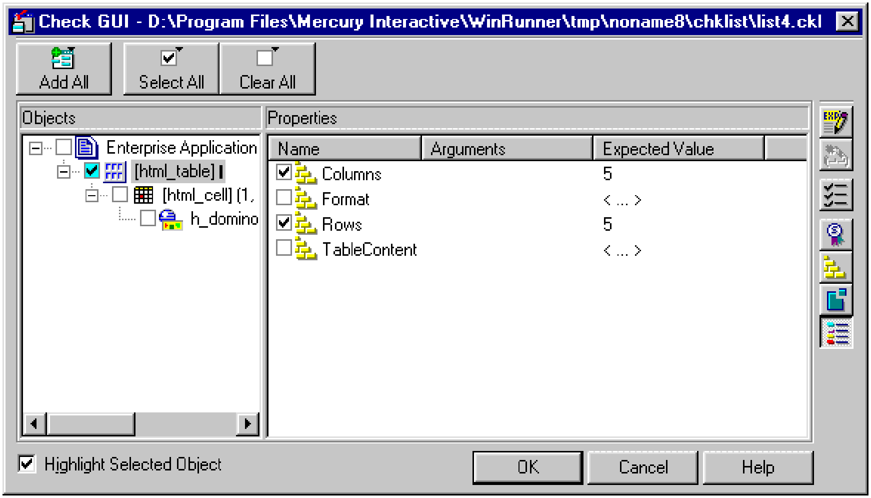
You can create a checkpoint to check the number of columns and rows in a table.

To check the number of columns and rows in a table:

1 Choose **Create > GUI Checkpoint > For Object/Window.**

The WinRunner window is minimized to an icon, the mouse pointer turns into a pointing hand, and a help window opens.

2 Double-click a table on your Web page. The **Check GUI** dialog box opens, and the object is highlighted.



Name	Arguments	Expected Value
☑ Columns		5
□ Format		< ... >
☑ Rows		5
□ TableContent		< ... >

☑ Highlight Selected Object

Books Online
Find
Find Again
Help
Top of Chapter
Back

3 In the **Objects** column, make sure the table is selected.

The Properties column indicates the properties available for you to check.

4 In the **Properties** column, select the **Columns** and/or **Rows** check box.

5 To edit the expected value of a property, click **Columns** or **Rows**. The property is highlighted.

6 Click the **Edit Expected Value** button, or double-click the value in the **Expected Value** column to edit it. A spin box opens.

7 Edit the expected value of the property, as desired.

8 Click **OK** to close the Check GUI dialog box.

WinRunner captures the object information and stores it in the test's expected results folder. The WinRunner window is restored and a checkpoint appears in your test script as an **obj_check_gui** or **win_check_gui** statement. For more information on **obj_check_gui** and **win_check_gui** functions, refer to the *TSL Online Reference* (**Help > TSL Online Reference**).

Books Online
Find
Find Again
Help
Top of Chapter
Back

Checking the URL of Links

You can use a checkpoint to check the URL of a text link or an image link in your Web page.

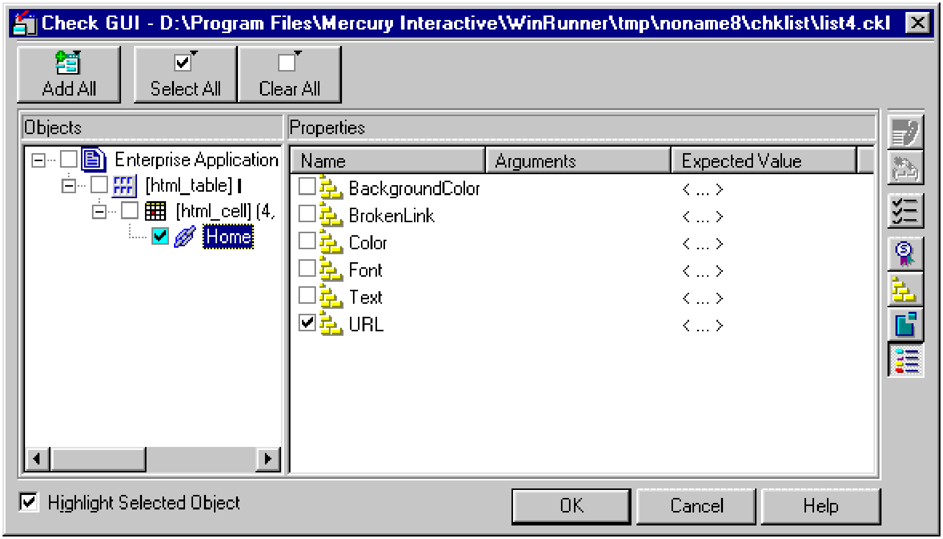
To check the URL of a link:

1 Choose **Create > GUI Checkpoint > For Object/Window**.

The WinRunner window is minimized to an icon, the mouse pointer turns into a pointing hand, and a help window opens.

Books Online
Find
Find Again
Help
Top of Chapter
Back

2 Double-click a text link on your Web page. The **Check GUI** dialog box opens, and the object is highlighted.



Check GUI - D:\Program Files\Mercury Interactive\WinRunner\tmp\noname8\chklist\list4.ckl
Add All
Select All
Clear All
Objects
Properties
OK
Cancel
Help

Objects:

- □ Enterprise Application
  - □ [html_table] I
    - □ [html_cell] (4,
      - ☑ Home

Name	Arguments	Expected Value
□ BackgroundColor		< ... >
□ BrokenLink		< ... >
□ Color		< ... >
□ Font		< ... >
□ Text		< ... >
☑ URL		< ... >

☑ Highlight Selected Object

3 In the **Objects** column, make sure that link is selected.

The **Properties** column indicates the properties available for you to check.

4 In the **Properties** column, select **URL** to check address of the link.

5 To edit the expected value of a the URL property, click **URL.** The property is highlighted.

Books Online
Find
Find Again
Help
Top of Chapter
Back

6. Click the **Edit Expected Value** button, or double-click the value in the **Expected Value** column to edit it. An edit box opens.
7. Edit the expected value.
8. Click **OK** to close the Check GUI dialog box.

   WinRunner captures the object information and stores it in the test's expected results folder. The WinRunner window is restored and a checkpoint appears in your test script as **obj_check_gui** statement. For more information on **obj_check_gui** functions, refer to the *TSL Online Reference* (**Help > TSL Online Reference**).

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Checking Source or Type of Images and Image Links

You can use a checkpoint to check the source and the image type of an image or an image link in your Web page.

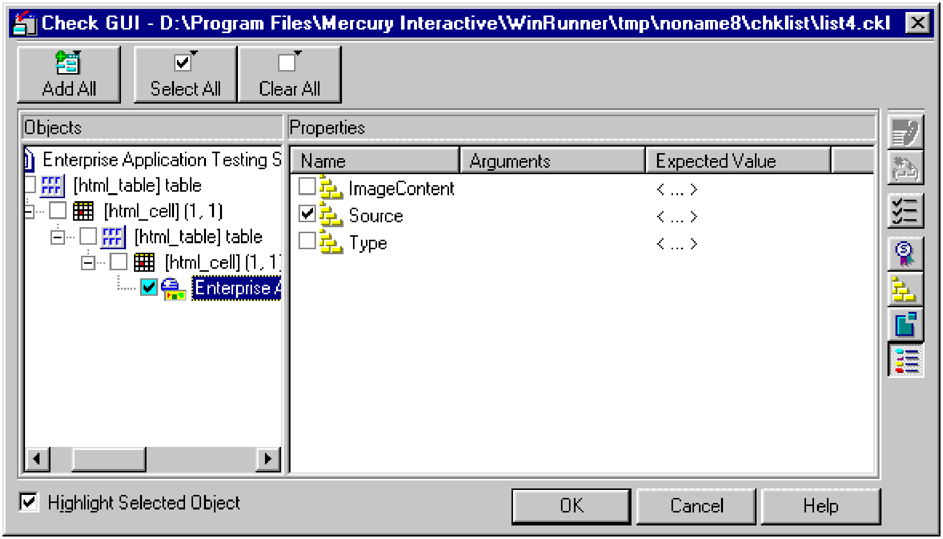
To check the source or type of an image or an image link:

1 Choose **Create > GUI Checkpoint > For Object/Window**.

The WinRunner window is minimized to an icon, the mouse pointer turns into a pointing hand, and a help window opens.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
⇦ Back

2 Double-click an image or image link on your Web page. The **Check GUI** dialog box opens, and the object is highlighted.



Check GUI - D:\Program Files\Mercury Interactive\WinRunner\tmp\noname8\chklist\list4.ckl

Enterprise Application Testing S
□ [html_table] table
□ [html_cell] (1, 1)
□ [html_table] table
□ [html_cell] (1, 1
☑ Enterprise A

Name	Arguments	Expected Value
□ ImageContent		< ... >
☑ Source		< ... >
□ Type		< ... >

☑ Highlight Selected Object

3 In the **Objects** column, make sure that the image or the image link is selected.

The Properties column indicates the properties available for you to check.

Books Online
Find
Find Again
Help
Top of Chapter
Back

4 In the Properties column, select a property check.

- **Source** indicates the location of the image.
- **Type** indicates whether the object is a plain image, an image link, or an image map.

5 To edit the expected value of a the property, click a property. The property is highlighted.

6 Click the **Edit Expected Value** button, or double-click the value in the **Expected Value** column to edit it. An edit box opens.

7 Edit the expected value and press **Enter**.

8 Click **OK** to close the Check GUI dialog box.

WinRunner captures the object information and stores it in the test's expected results directory. The WinRunner window is restored and a checkpoint appears in your test script as **obj_check_gui** statement. For more information on **obj_check_gui** functions, refer to the *TSL Online Reference* (**Help > TSL Online Reference**).

Books Online
Find
Find Again
Help
Top of Chapter
Back

Checking Color or Font of Text Links

You can use a checkpoint to check the color and font of a text link in your Web page.

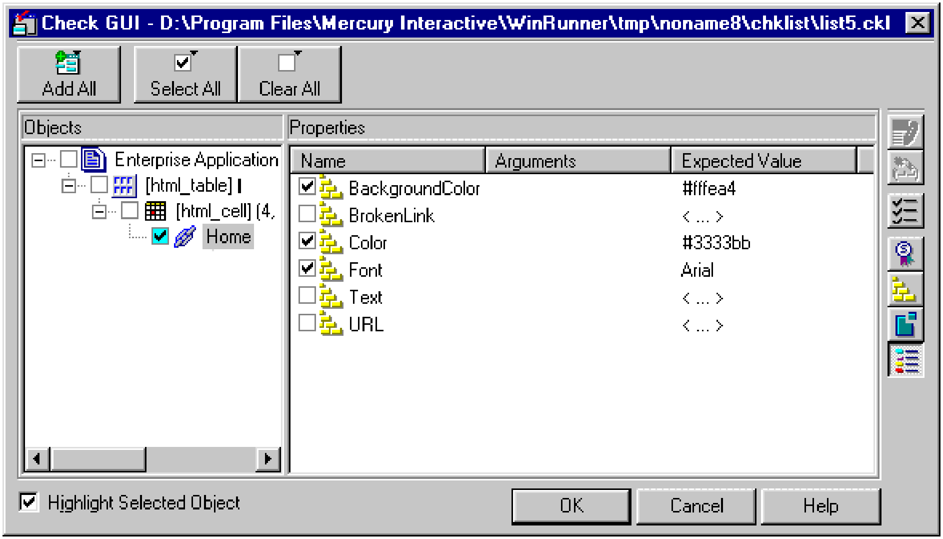
To check the color or font of a text link:

1 Choose **Create > GUI Checkpoint > For Object/Window**.

The WinRunner window is minimized to an icon, the mouse pointer turns into a pointing hand, and a help window opens.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

2 Double-click a text link on your Web page. The **Check GUI** dialog box opens, and the object is highlighted.



Check GUI - D:\Program Files\Mercury Interactive\WinRunner\tmp\noname8\chklist\list5.ckl

Objects:

- □ Enterprise Application
  - □ [html_table] I
    - □ [html_cell] (4,
      - ☑ Home

	Name	Arguments	Expected Value
☑	BackgroundColor		#fffea4
□	BrokenLink		< ... >
☑	Color		#3333bb
☑	Font		Arial
□	Text		< ... >
□	URL		< ... >

☑ Highlight Selected Object

3 In the **Objects** column, make sure that the text link is selected.

The Properties column indicates the properties available for you to check.

Books Online
Find
Find Again
Help
Top of Chapter
Back

4 In the Properties column, select a property check.

- **BackgroundColor** indicates the background color of a text link.
- **Color** indicates the foreground color of a text link.
- **Font** indicates the font of a text link.

5 To edit the expected value of a property, click a property. The property is highlighted.

6 Click the **Edit Expected Value** button, or double-click the value in the **Expected Value** column to edit it. A box opens.

7 Edit the expected value and press **Enter**.

8 Click **OK** to close the Check GUI dialog box.

WinRunner captures the object information and stores it in the test's expected results directory. The WinRunner window is restored and a checkpoint appears in your test script as **obj_check_gui** statement. For more information on **obj_check_gui** functions, refer to the *TSL Online Reference* (**Help > TSL Online Reference**).

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Checking Text

You can check text in a Web object or in any area of your Web page.

This chapter describes:

- **Reading All the Text in a Frame or an Object**
- **Reading a Text String from a Frame or an Object**
- **Checking that a Text String Exists in a Frame or an Object**

Books Online
Find
Find Again
Help
Top of Chapter
Back

About Checking Text

You can use text checkpoints in your test scripts to read and check text in Web objects and in areas of the Web page. While creating a test, you point to an object or a frame containing text. WebTest reads the text and writes a TSL statement to the test script. You may then add simple programming elements to your test scripts to verify the contents of the text.

You can use a text checkpoint to:

- read a text string or all the text from a Web object or frame, using **web_obj_get_text** or **web_frame_get_text**
- check that a text string exists in a Web object or frame, using **web_obj_text_exists** or **web_frame_text_exists**

Books Online
Find
Find Again
Help
Top of Chapter
Back

Reading All the Text in a Frame or an Object

You can read all the visible text in a frame or an object using **web_obj_get_text** or **web_frame_get_text**.

To read all the text in a frame or an object:

1 Choose **Create > Get Text > From Object/Window.**

WinRunner is minimized, the mouse pointer becomes a pointing hand, and a Help window opens.

2 Click the Web object or the frame.

WinRunner captures the text in the object and a **web_obj_get_text** or a **web_frame_get_text** statement is inserted in your test script.

Note: When the WebTest add-in is not loaded, or when a non-Web object is selected, WinRunner generates a **win_get_text** or **obj_get_text** statement in your test script. For more information on the **_get_text** functions, refer to the *WinRunner User’s Guide* and the *TSL Online Reference*.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Reading a Text String from a Frame or an Object

You can read a text string from a frame or an object using **web_obj_get_text** or **web_frame_get_text**.

To read a text string from a frame or an object:

1 Choose **Create > Get Text > From Selection (WebTest only)**.

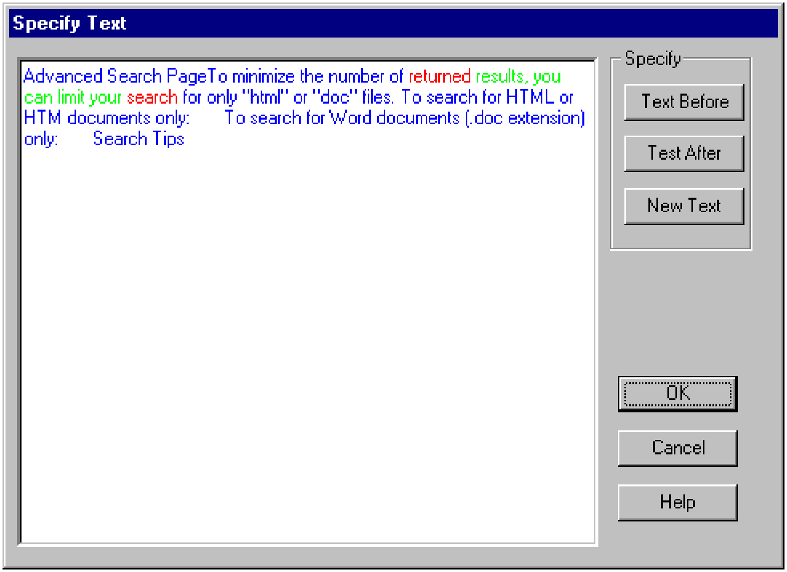
WinRunner is minimized, the mouse pointer becomes a pointing hand, and a Help window opens.

2 Highlight the text string to be read.

3 On the highlighted text string, right-click the mouse button to capture the string.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

The **Specify Text** dialog box opens. The text string to be read appears in green. The red text that appears on the left and right of your selection, defines the bounds of the string.



Advanced Search PageTo minimize the number of returned results, you can limit your search for only "html" or "doc" files. To search for HTML or HTM documents only: To search for Word documents (.doc extension) only: Search Tips

Books Online
Find
Find Again
Help
Top of Chapter
Back

4. You can modify your text selections.

   - To modify your highlighted text selection, highlight a new text string and click **New Text**.

     Your new text selection appears in green. The text that appears before and after your text string appears in red.

   - To modify the red text string that appears to the left of your selection, highlight a new text string and click **Text Before**.

   - To modify the red text string that appears to the right of your selection, highlight a new text string and click **Text After**.

5. Click **OK** to close the Specify Text dialog box.

   The WinRunner window is restored and a **web_obj_get_text** or a **web_frame_get_text** statement is inserted in your test script.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Checking that a Text String Exists in a Frame or an Object

You can check whether a text string exists in an object or a frame using **web_obj_text_exists** or **web_frame_text_exists**.

To check that a text string exists in a frame or an object:

1 Choose **Create > Get Text > WebTest Checkpoint**.

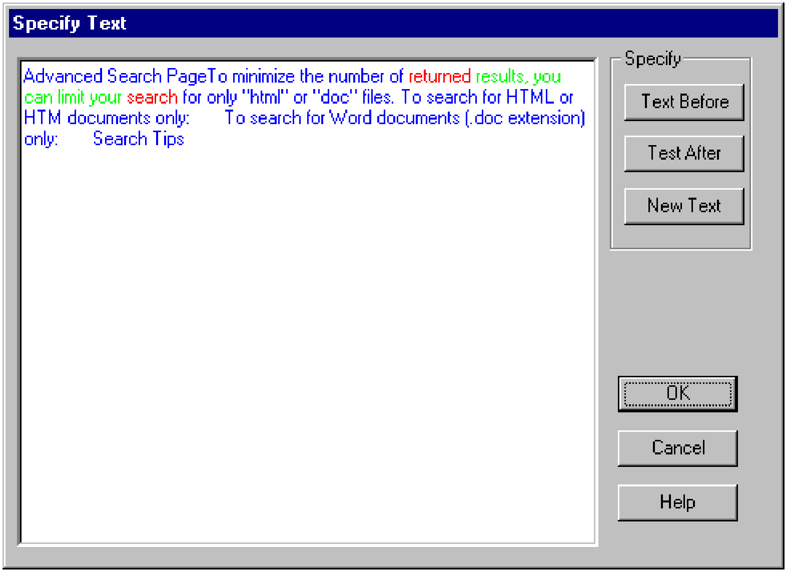
WinRunner is minimized, the mouse pointer becomes a pointing hand, and a Help window opens.

2 Highlight the text string to be checked.

3 On the highlighted text string, right-click the mouse button to capture the string.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

The **Specify Text** dialog box opens. The text string to be checked appears in green. The red text that appears on the left and right of your selection, defines the bounds of the string.



Advanced Search PageTo minimize the number of returned results, you can limit your search for only "html" or "doc" files. To search for HTML or HTM documents only: To search for Word documents (.doc extension) only: Search Tips

4 You can modify your text selections.

- To modify your highlighted text selection, highlight a new text string and click **New Text**.

  Your new text selection appears in green. The text that appears before and after your text string appears in red.

- To modify the red text string that appears to the left of your selection, highlight a new text string and click **Text Before**.
- To modify the red text string that appears to the right of your selection, highlight a new text string and click **Text After**.

5 Click **OK** to close the Specify Text dialog box.

The WinRunner window is restored and a **web_obj_text_exists** or a **web_frame_text_exists** statement is inserted in your test script.

Note: After you run your test, a *check text* statement appears in the test log of your Test Results window.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Handling Exceptions

You can instruct WebTest to handle unexpected events and errors that occur in your testing environment during a test run.

This chapter describes:

- Changing the Status of an Exception
- Modifying an Exception
- Adding a New Exception

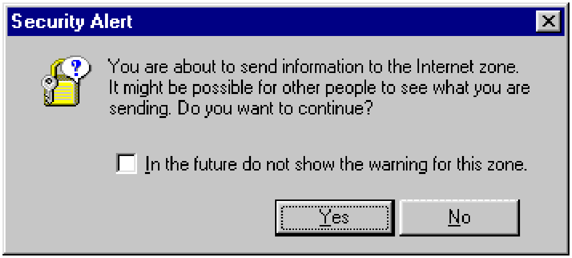
Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

About Handling Exceptions

Unexpected events and errors during a test run can disrupt your test and distort test results. This is a problem particularly when running tests unattended: the tests are suspended until you perform the action needed to recover.

Using the *Web Exception Editor,* you can instruct WebTest to detect and handle the appearance of a specific dialog box and act to recover the test run.

For example, if a Security Alert dialog box appears during a test run, you can instruct WebTest to recover the test run, by clicking the Yes button.



Security Alert

You are about to send information to the Internet zone. It might be possible for other people to see what you are sending. Do you want to continue?

□ In the future do not show the warning for this zone.

Books Online
Find
Find Again
Help
Top of Chapter
Back

The Web Exception Editor contains a list of exceptions that WebTest supports. You can modify the list and configure additional types of dialog box exceptions that you would like WebTest to support.

You can:

- change the status of an exception
- modify the description of an exception
- add a new exception.

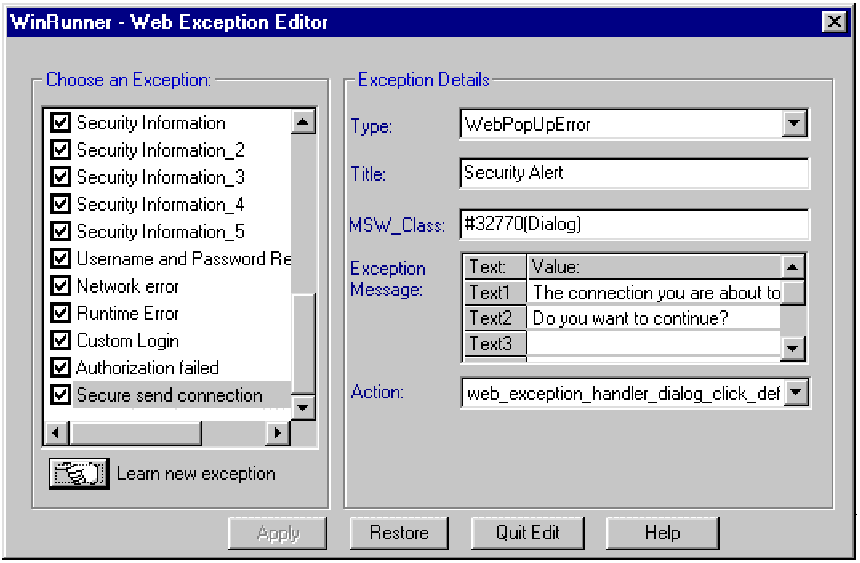
Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Changing the Status of an Exception

The Web Exception Editor includes a list of all the available exceptions. You can choose to activate or deactivate any exception in the list.

To change the status of an exception:

1 Choose **Tools > Web Exception Handling**. The **Web Exception Editor** opens.



WinRunner - Web Exception Editor

Choose an Exception:

- ☑ Security Information
- ☑ Security Information_2
- ☑ Security Information_3
- ☑ Security Information_4
- ☑ Security Information_5
- ☑ Username and Password Re
- ☑ Network error
- ☑ Runtime Error
- ☑ Custom Login
- ☑ Authorization failed
- ☑ Secure send connection

Exception Details

Type: WebPopUpError

Title: Security Alert

MSW_Class: #32770(Dialog)

Exception Message:

Text:	Value:
Text1	The connection you are about to
Text2	Do you want to continue?
Text3	

Action: web_exception_handler_dialog_click_def

2 In the **Choose an Exception** list, click an exception.

Books Online
Find
Find Again
Help
Top of Chapter
Back

The exception is highlighted. The current description of the exception appears in the Exception Details area.

3. To activate an exception, select its check box. To deactivate the exception, clear its check box.
4. Click **Apply**. The **Save Configuration** dialog box opens.
5. Click **OK** to save the changes to the configuration file.
6. Click **Quit Edit** to exit the Web Exception Editor.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

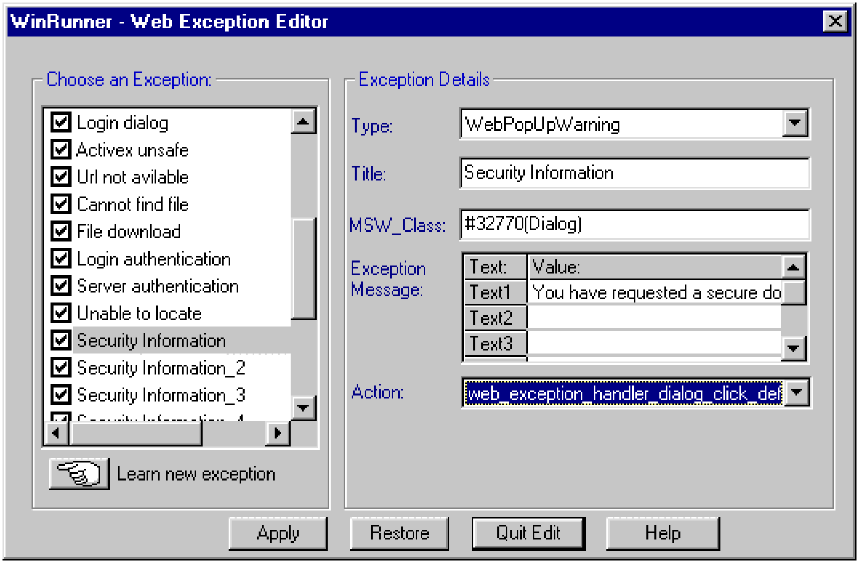
Modifying an Exception

You can modify the details of an exception listed in the Web Exception Editor.

To modify the details of an exception:

1 Choose **Tools > Web Exception Handling**. The **Web Exception Editor** opens.

2 In the **Choose an Exception list**, click an exception.



Choose an Exception:

- ☑ Login dialog
- ☑ Activex unsafe
- ☑ Url not avilable
- ☑ Cannot find file
- ☑ File download
- ☑ Login authentication
- ☑ Server authentication
- ☑ Unable to locate
- ☑ Security Information
- ☑ Security Information_2
- ☑ Security Information_3
- ☑ Security Information_4

Type: WebPopUpWarning

Title: Security Information

MSW_Class: #32770(Dialog)

Exception Message:

Text:	Value:
Text1	You have requested a secure do
Text2	
Text3	

Action: web_exception_handler_dialog_click_def

Books Online
Find
Find Again
Help
Top of Chapter
Back

The exception is highlighted. The current description of the exception appears in the Exception Details area.

3 To modify the exception type, choose a type from the **Type** list.

- **WebPopUpError** indicates that the exception is an error dialog box
- **WebPopUpWarning** indicates that the exception is a warning dialog box.

4 To modify the title of the dialog box, type a new title in the **Title** box.

5 To modify the text that appears in the exception dialog box, click a text line and edit the text.

6 To change the action that is responsible for recovering test execution, choose an action from the **Action** list.

- **Web_exception_handler_dialog_click_default** activates the default button.
- **Web_exception_handler_fail_retry** activates the default button and reloads the Web page.
- **Web_exception_enter_username_password** uses the given user name and password.
- **Web_exception_handler_dialog_click_yes** activates the Yes button.
- **Web_exception_handler_dialog_click_no** activates the No button.

7 Click **Apply.** The **Save Configuration** dialog box opens.

8 Click **OK** to save the changes to the configuration file.

9 Click **Quit Edit** to exit the Web Exception Editor.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Adding a New Exception

You can add a new exception to the list of exceptions in the Web Exception Editor.

To add a new exception:

1. Choose **Tools > Web Exception Handling**. The **Web Exception Editor** opens.
2. Click the pointing hand and click the dialog box. A new exception is added to the list.
3. In the **Type** list, select an exception type.
   - **WebPopUpError** indicates that the exception is an error dialog box
   - **WebPopUpWarning** indicates that the exception is a warning dialog box.

The Editor displays the title, MSW_Class, and message of the exception.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

4. In the **Action** list, choose the handler function action that is responsible for recovering test execution.

   - **Web_exception_handler_dialog_click_default** activates the default button.
   - **Web_exception_handler_fail_retry** activates the default button and reloads the Web page.
   - **Web_exception_enter_username_password** uses the given user name and password.
   - **Web_exception_handler_dialog_click_yes** activates the Yes button.
   - **Web_exception_handler_dialog_click_no** activates the No button.

5. Click **Apply.** The **Save Configuration** dialog box opens.
6. Click **OK** to save the changes to the configuration file.
7. Click **Quit Edit** to exit the Web Exception Editor.

Books Online
Find
Find Again
Help
Top of Chapter
Back

Once you have created a test script, you run the test to check the behavior of your Web site.

About Running Tests

When you run a test, WinRunner interprets your test script, line by line, and performs the operations on your Web site.

Use WinRunner's Run commands to run your tests. You can run an entire test or a portion of a test. For more information, refer to the *WinRunner User's Guide*.

Note: If you created permanent GUI map files for you tests, you must load the appropriate GUI map files before you run your tests. For more information, refer to Chapter 4, "Creating the GUI Map", in the *WinRunner User's Guide*.

Books Online
Find
Find Again
Help
Top of Chapter
Back

Running a Test to Check Your Web Site

When you run a test to check the behavior of your Web site, WinRunner compares the current results with the expected results. You specify the folder in which the verification results for the test are saved.

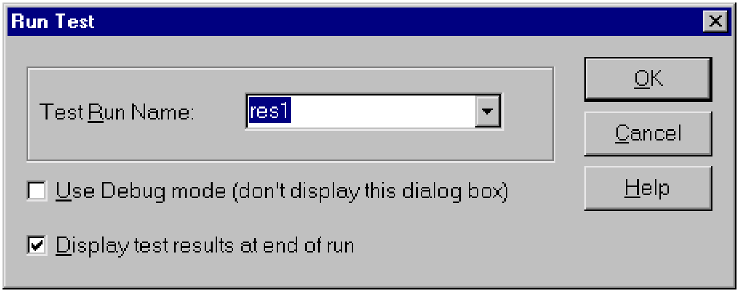
To run a test to check your Web site:

1 Open the test if it is not already open.

2 Make sure that **Verify** is selected from the dropdown list of run modes on the toolbar.

3 Choose **Run > Run from Top**, or click the **Run from Top** button.

4 In the Run Test dialog box, assign a name to the folder that will store the results, or accept the default name "res1".



Test Run Name: res1

□ Use Debug mode (don't display this dialog box)

☑ Display test results at end of run

Books Online
Find
Find Again
Help
Top of Chapter
Back

5 To instruct WinRunner to display the test results automatically following the test run (the default), select the **Display test results at end of run** check box.

6 Click **OK**. The Run Test dialog box closes and WinRunner runs the test.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Analyzing Test Results

After you execute a test, you can view a report of all the major events that occurred during the test run in order to determine its success or failure.

This chapter describes:

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

About Analyzing Test Results

When a test run is completed, you can view detailed test results in the WinRunner Test Results window. The window contains a description of the major events that occurred during the test run, such as errors and checkpoints. You can view expected, debug, and verification results in the Test Results window. By default, the Test Results window displays the results of the most recently executed test run. For more information, refer to the *WinRunner User’s Guide.*

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Viewing Results of a Test Run

When a test run is completed, test results are displayed in the WinRunner Test Results window.

To view test results:

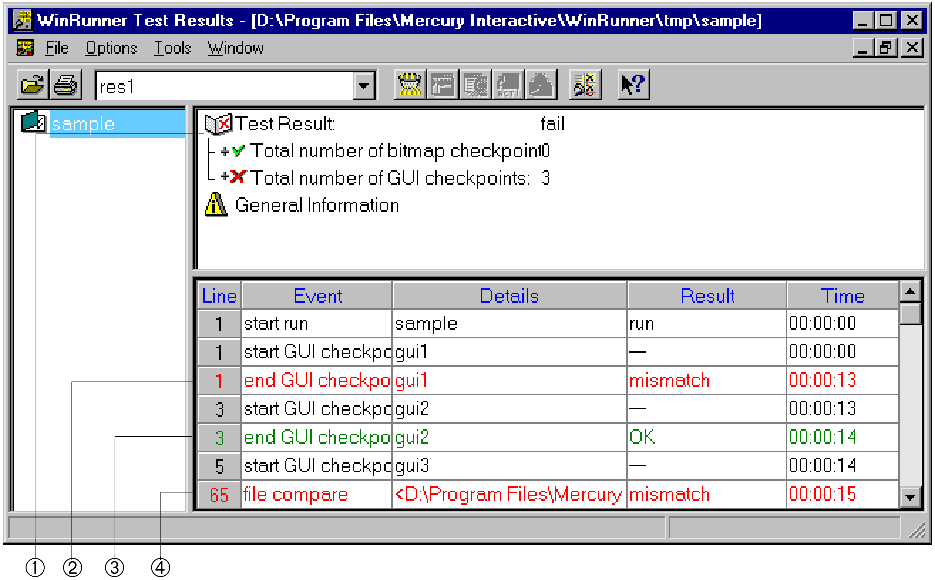
1 To open the WinRunner Test Results window, choose **Tools > Test Results** or click the **Test Results** button. Note, if the **Display test results at end of run** check box was selected in the Run Test dialog box before you ran the test, the **Test Results** window opens automatically.

1 *Indicates whether the test passed or failed, and lists all checkpoints performed during the test run.*

2 *Failed GUI checkpoint.*

3 *Successful checkpoint.*

4 *Checkpoint that verified the content of a web page.*


WinRunner Test Results - [D:\Program Files\Mercury Interactive\WinRunner\tmp\sample]
File Options Tools Window
res1
sample
①
②
③
④

Test Result: fail
+✓ Total number of bitmap checkpoint0
+✗ Total number of GUI checkpoints: 3
General Information

Line	Event	Details	Result	Time
1	start run	sample	run	00:00:00
1	start GUI checkpo	gui1	—	00:00:00
1	end GUI checkpo	gui1	mismatch	00:00:13
3	start GUI checkpo	gui2	—	00:00:13
3	end GUI checkpo	gui2	OK	00:00:14
5	start GUI checkpo	gui3	—	00:00:14
65	file compare	<D:\Program Files\Mercury	mismatch	00:00:15

Books Online
Find
Find Again
Help
Top of Chapter
Back

- In the Test Results section you can see whether the test passed or failed, and how many checkpoints were included in the test.
- In the test log, look for GUI check statements. Failed checkpoints appear in red; passed checkpoints appear in green.
- If you created a checkpoint to verify the text content of a cell or a frame, a file compare statement will appear above that checkpoint statement.

2 By default, the Test Results window displays the results of the most recently executed test run.

To view other test run results, click the **Results** box in the toolbar and select a test run.

3 To view a text version of the test results, choose **Tools > Text Report** from the Test Results window; the report opens in Notepad.

4 To view only specific types of results in the events column in the test log, choose **Options > Filters** or click the **Filters** button.

5 To print test results directly from the Test Results window, choose **File > Print** or click the **Print** button.

In the Print dialog box, choose the number of copies you want to print and click OK. Test results print in a text format.

6 To close the Test Results window, choose **File > Exit**.

Books Online
Find
Find Again
Help
Top of Chapter
Back

Viewing Checkpoint Results

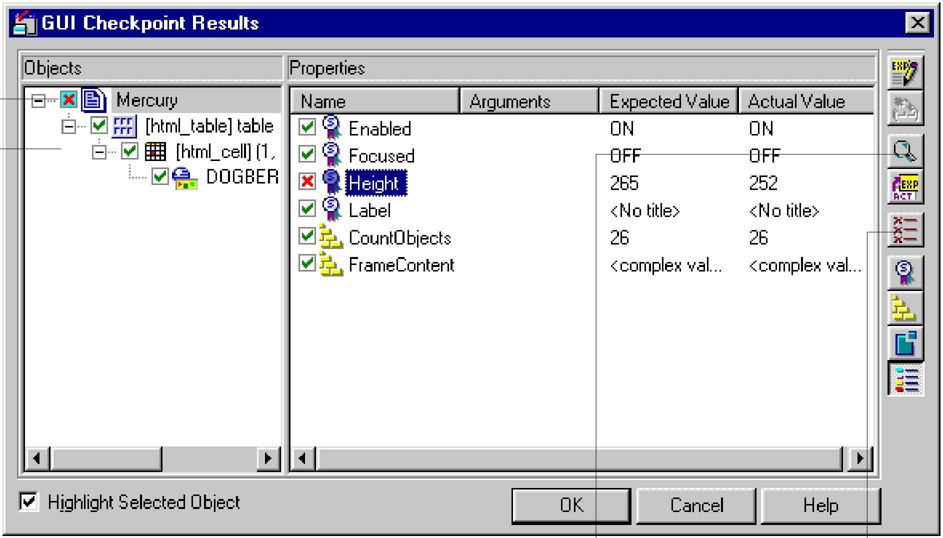
You can view the results of a checkpoint using the GUI Checkpoint Results dialog box.

To view the results of a checkpoint:

1 Open the Test Results window. In the test log, look for an **end GUI checkpoint** entry in the Event column.

Books Online
Find
Find Again
Help
Top of Chapter
Back

2 Double-click an **end GUI checkpoint** entry in the Event column. The **GUI Checkpoint Results** dialog box opens.



- ☒ Mercury
  - ☑ [html_table] table
    - ☑ [html_cell] (1,
      - ☑ DOGBER

Name	Arguments	Expected Value	Actual Value
☑ Enabled		ON	ON
☑ Focused		OFF	OFF
☒ Height		265	252
☑ Label		<No title>	<No title>
☑ CountObjects		26	26
☑ FrameContent		<complex val...	<complex val...

☑ Highlight Selected Object

Failed check

Passed check

Compare expected and actual values

Show failures only

The Objects column lists all the objects that WinRunner checked in the specified Web page. The checked objects are indented. The Properties columns lists the properties of the Web object that WinRunner checked.

A green mark indicates a passed check. A red mark indicates a failed check.

Books Online
Find
Find Again
Help
Top of Chapter
Back

3. In the **Objects** column, select an object.

   The **Properties** column lists the object properties checked and the results of the check.

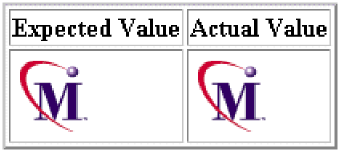
   - **Name** indicates the name of each object property that WinRunner checked.
   - **Arguments** indicates the specified arguments.
   - **Expected Value** indicates the expected value of object property that WinRunner checked.
   - **Actual Value** indicates the actual value of each object property that WinRunner checked.

4. In the **Properties** column, select a property and click the **Compare Expected and Actual Values** button.

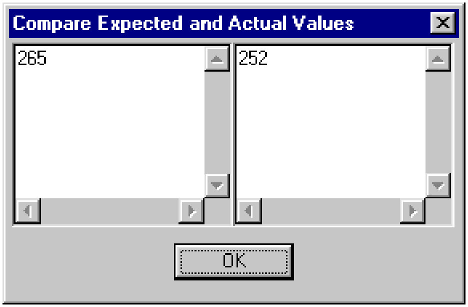
   - When viewing the results of a text content check or a structure check on a frame or a cell, the **WDiff** utility opens. For more information see, **Understanding the WDiff Utility** on page 105.
   - When viewing the results of a text content check or a structure check of a table, the **Data Comparison Viewer** opens. For more information see, **Understanding the Data Comparison Viewer** on page 108.

Books Online
Find
Find Again
Help
Top of Chapter
Back

- When viewing the content check results of an image, the **Microsoft Internet Explorer** browser opens. It displays the expected and actual image.



- For all other properties, the **Compare Expected and Actual Values** dialog box opens. It displays the expected and actual values in separate columns.



Compare Expected and Actual Values

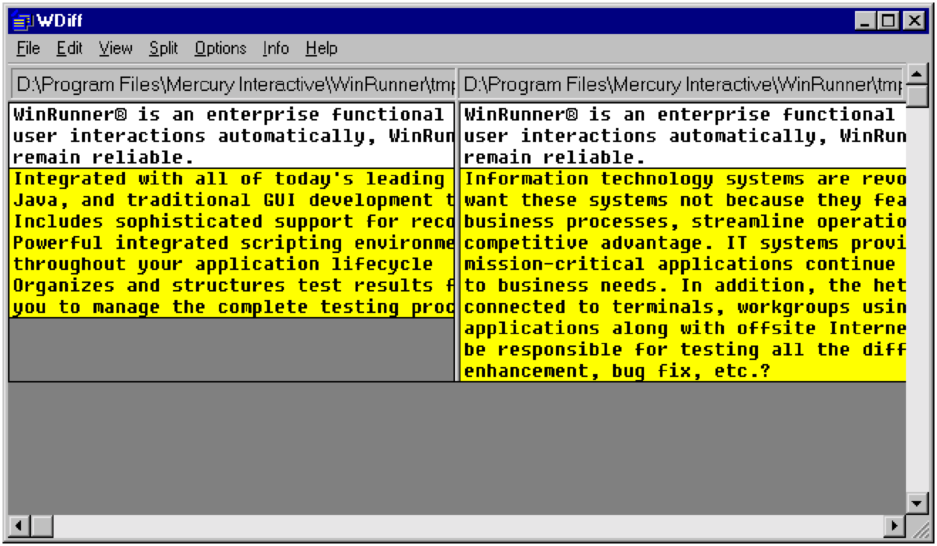
265

252

5 Click **OK** to close the GUI Checkpoint Results dialog box.

Understanding the WDiff Utility

You can view the results of text content checks or structure checks of cells or frames using the WDiff utility. The WDiff utility displays the expected and actual results. Differences are highlighted. This utility is accessed from the GUI Checkpoint Results dialog box. For more information on the GUI Checkpoint Results dialog box, see **Viewing Checkpoint Results** on page 101.



D:\Program Files\Mercury Interactive\WinRunner\tmp

```
WinRunner® is an enterprise functional
user interactions automatically, WinRun
remain reliable.
Integrated with all of today's leading
Java, and traditional GUI development t
Includes sophisticated support for reco
Powerful integrated scripting environme
throughout your application lifecycle
Organizes and structures test results f
you to manage the complete testing proc
```

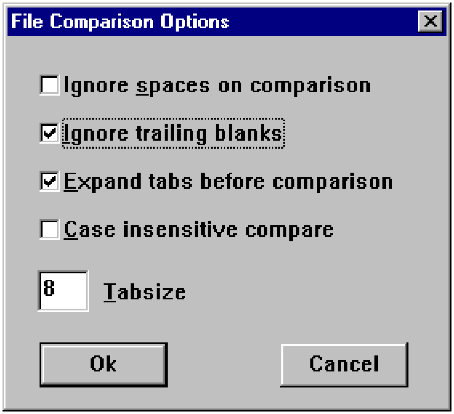
D:\Program Files\Mercury Interactive\WinRunner\tmp

```
WinRunner® is an enterprise functional
user interactions automatically, WinRun
remain reliable.
Information technology systems are revo
want these systems not because they fea
business processes, streamline operatio
competitive advantage. IT systems provi
mission-critical applications continue
to business needs. In addition, the het
connected to terminals, workgroups usin
applications along with offsite Interne
be responsible for testing all the diff
enhancement, bug fix, etc.?
```

- To see the next mismatch in a file, choose **View > Next Difference**, or press **Tab**. The window scrolls to the next highlighted line. To see the previous difference, choose **View > Previous Difference** or press the **Backspace** key.

Books Online
Find
Find Again
Help
Top of Chapter
Back

- To view only the lines in the files that contain mismatches, choose **Options > View > Hide Matching Areas**, (a check mark at the left side of the menu identifies the current state). The window shows only the highlighted parts of both files.
- To modify the way the actual and expected results are compared, choose **Options > File Comparison**.



File Comparison Options

☐ Ignore spaces on comparison

☑ Ignore trailing blanks

☑ Expand tabs before comparison

☐ Case insensitive compare

8 Tabsize

Note that when you modify any of the options, the two files are read and compared again.

- **Ignore spaces on comparison:** Tab characters and spaces are ignored on comparison.

Books Online
Find
Find Again
Help
Top of Chapter
Back

- **Ignore trailing blanks (default):** One or more blanks at the end of a line are ignored during the comparison.
- **Expand tabs before comparison (default):** Tab characters (hex 09) in the text are expanded to the number of spaces which are necessary to reach the next tab stop. The number of spaces between tab stops is specified in the **Tabsize** parameter. This **expand tabs before comparison** option will be ignored, if the **Ignore spaces on comparison** option is selected at the same time.
- **Case insensitive compare:** Uppercase and lowercase is ignored during comparison of the files.
- **Tabsize:** The tabsize (number of spaces between tab stops) is selected between 1 and 19 spaces. The default size is 8 spaces. The option influences the file comparison, if the **expand tabs before comparison** option is also set. Tabs are always expanded to the given number of spaces.

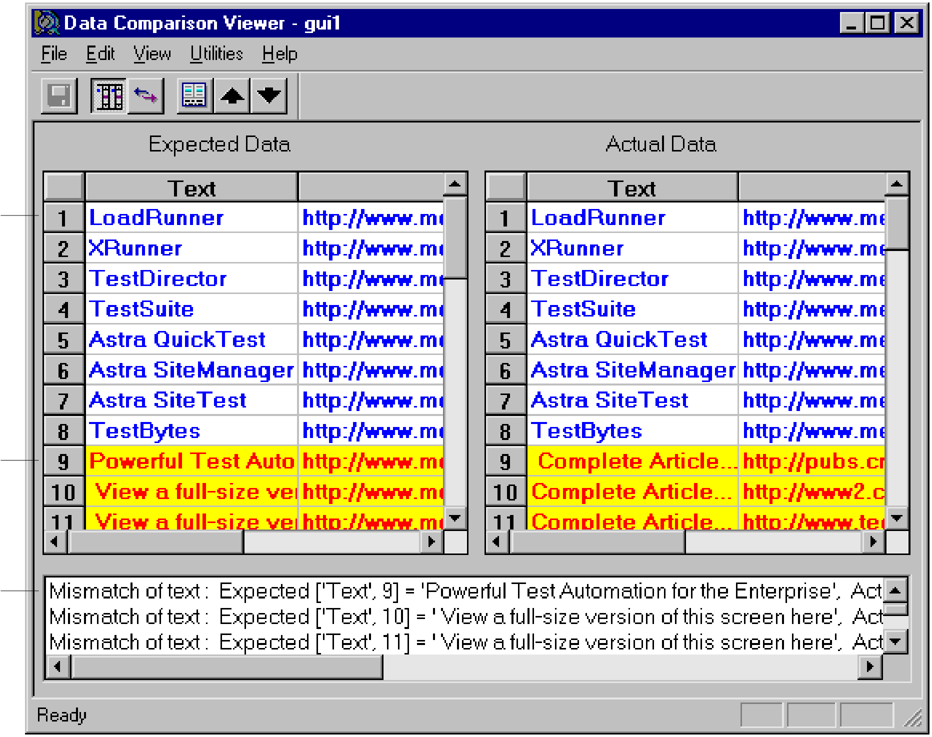
Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

Understanding the Data Comparison Viewer

You can view the results of checkpoints on links, images in frames, or table contents using the Data Comparison Viewer. The viewer displays the expected and actual results. Differences are highlighted. This viewer is accessed from the GUI Checkpoint Results dialog box. For more information on the GUI Checkpoint Results dialog box, see **Viewing Checkpoint Results** on page 101.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

When the Data Comparison Viewer opens, it shows both expected and actual results. All cells are color coded, and all errors and mismatches are listed at the bottom of the viewer.



Expected Data

	Text	
1	LoadRunner	http://www.m
2	XRunner	http://www.m
3	TestDirector	http://www.m
4	TestSuite	http://www.m
5	Astra QuickTest	http://www.m
6	Astra SiteManager	http://www.m
7	Astra SiteTest	http://www.m
8	TestBytes	http://www.m
9	Powerful Test Auto	http://www.m
10	View a full-size ve	http://www.m
11	View a full-size ve	http://www.m

Actual Data

	Text	
1	LoadRunner	http://www.me
2	XRunner	http://www.me
3	TestDirector	http://www.me
4	TestSuite	http://www.me
5	Astra QuickTest	http://www.me
6	Astra SiteManager	http://www.me
7	Astra SiteTest	http://www.me
8	TestBytes	http://www.me
9	Complete Article...	http://pubs.c
10	Complete Article...	http://www2.c
11	Complete Article...	http://www.te

```
Mismatch of text :  Expected ['Text', 9] = 'Powerful Test Automation for the Enterprise',  Act
Mismatch of text :  Expected ['Text', 10] = ' View a full-size version of this screen here',  Act
Mismatch of text :  Expected ['Text', 11] = ' View a full-size version of this screen here',  Act
```

Cell does not contain a mismatch

Cell contains a mismatch

List of errors and mismatches

Books Online
Find
Find Again
Help
Top of Chapter
Back

Use the following color codes to interpret the differences that are highlighted in your window:

- **Blue on white background:** Cell was included in the comparison and no mismatch was found.
- **Cyan on ivory background:** Cell was not included in the comparison.
- **Red on yellow background:** Cell contains a mismatch.
- **Magenta on green background:** Cell was verified but not found in the corresponding table.
- **Background color only:** cell is empty (no text).

- By default, scrolling between the expected and actual tables in the Data Comparison Viewer is synchronized. When you click any cell, the corresponding cell in the other table flashes red.

  To scroll through the tables separately, clear **Synchronize Scrolling** from the **Utilities** menu. Use the scroll bar as needed to view hidden parts of the table.

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

- To filter a list of errors and mismatches that appear at the bottom of the Data Comparison Viewer, use the following options:
  - **To view mismatches for a specific column only:** Double-click a column heading (the column name) in either table.
  - **To view mismatches for a single row:** Double-click a row number in either table.
  - **To view mismatches for a single cell:** Double-click a cell with a mismatch.
  - **To see the full list of mismatches:** Click **Utilities > Full List** or double-click the empty cell in the upper left corner of the table.
  - **To clear the list:** Double-click a cell with no mismatch.
  - **To see the cell(s) that correspond to a listed mismatch:** Click a mismatch in the list at the bottom of the dialog box to see the corresponding cells in the table flash red. If the cell with the mismatch is not visible, the table scroll automatically to display it.

Index

Books Online
Find
Find Again
? Help
◀ ▶
Top of Chapter
⇦ Back

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
⇦ Back

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Books Online
Find
Find Again
Help
Top of Chapter
Back

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Books Online
Find
Find Again
Help
◀ ▶
Top of Chapter
Back

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

WebTest User’s Guide, Version 6.0

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

Mercury Interactive Corporation
1325 Borregas Avenue
Sunnyvale, CA 94089
Tel. (408)822-5200 (800) TEST-911
Fax. (408)822-5300

WRWTUG6.02/01

Books Online
Find
Find Again
Help
Back