EXPERIMENT 1

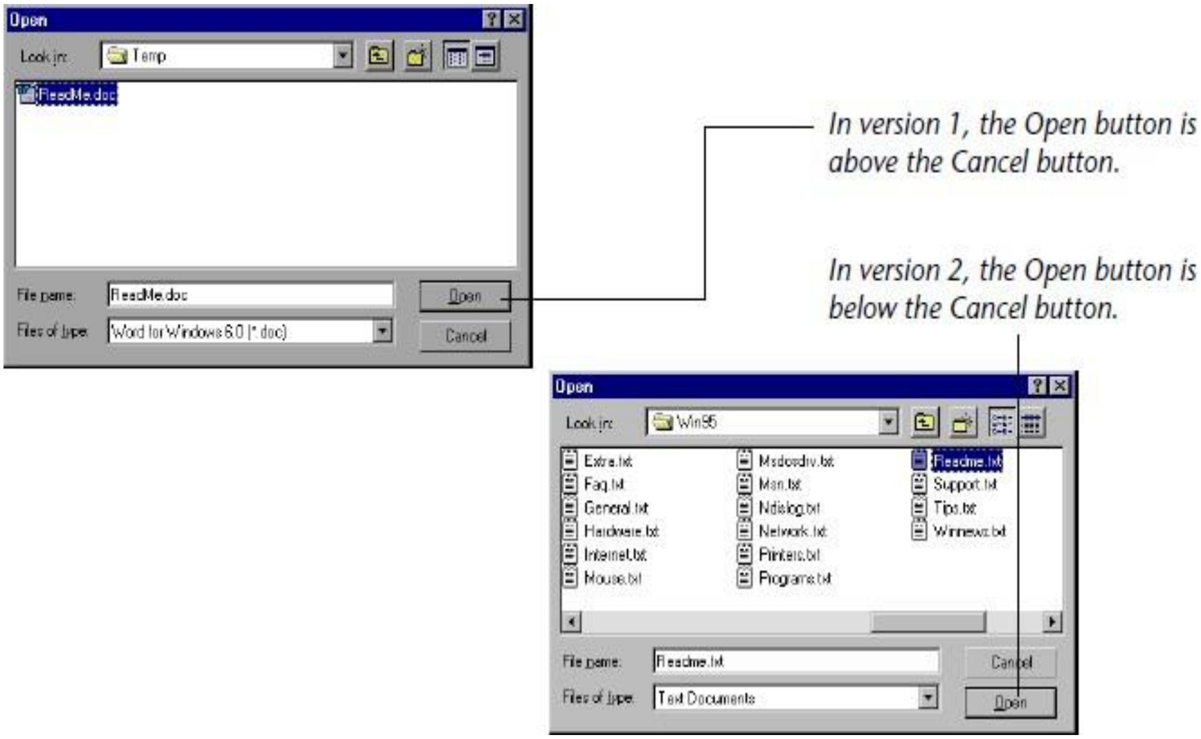
RECORDING IN CONTEXT SENSITIVE MODE AND ANALOG MODE

Context Sensitive mode records the operations you perform on your application in terms of its GUI objects. As you record, WinRunner identifies each GUI object you click (such as a window, button, or list), and the type of operation performed (such as drag, click, or select).

For example, if you click the **Open** button in an Open dialog box, WinRunner records the following:

button_press ("Open");

When it runs the test, WinRunner looks for the Open dialog box and the Open button represented in the test script. If, in subsequent runs of the test, the button is in a different location in the Open dialog box, WinRunner is still able to find it.



In version 1, the Open button is above the Cancel button.

In version 2, the Open button is below the Cancel button.

Use Context Sensitive mode to test your application by operating on its user interface.

To record a test in context sensitive mode:

1. Choose **Test** > **Record–Context Sensitive** or click the **Record–Context Sensitive** button.

   The letters Rec are displayed in dark blue text with a light blue background on the ***Record*** button to indicate that a context sensitive record session is active.

2. Perform the test as planned using the keyboard and mouse.

   Insert checkpoints and synchronization points as needed by choosing the appropriate commands from the User toolbar or from the **Insert** menu menu: GUI Checkpoint, Bitmap Checkpoint, Database Checkpoint, or Synchronization Point.

3. To stop recording, click **Test** > **Stop Recording** or click **Stop**.

EXPERIMENT 2

GUI CHECKPOINT FOR SINGLE PROPERTY

You can check a single property of a GUI object. For example, you can check whether a button is enabled or disabled or whether an item in a list is selected. To create a GUI checkpoint for a property value, use the Check Property dialog box to add one of the following functions to the test script:

button_check_info scroll_check_info

edit_check_info static_check_info

list_check_info win_check_info

obj_check_info

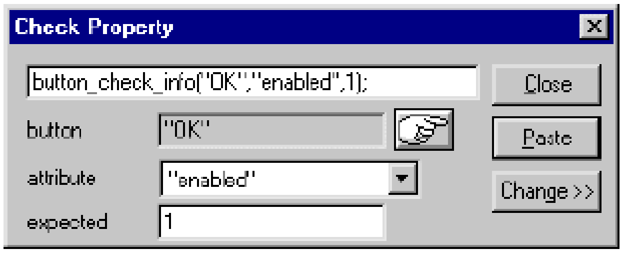
To create a GUI checkpoint for a property value:

1. Choose **Insert > GUI Checkpoint > For Single Property**. If you are recording in Analog mode, press the CHECK GUI FOR SINGLE PROPERTY softkey in order to avoid extraneous mouse movements.

   The WinRunner window is minimized, the mouse pointer becomes a pointing hand, and a help window opens on the screen.

2. Click an object.

   The Check Property dialog box opens and shows the default function for the selected object. WinRunner automatically assigns argument values to the function.



button_check_info("OK","enabled",1);

button "OK"

attribute "enabled"

expected 1

1. You can modify the arguments for the property check.

   - To modify assigned argument values, choose a value from the **Attribute** list. The expected value is updated in the Expected text box.
   - To choose a different object, click the pointing hand and then click an object in your application. WinRunner automatically assigns new argument values to the function.

   Note: that if you click an object that is not compatible with the selected function, a message states that the current function cannot be applied to the selected object. Click OK to clear the message, and then click Close to close the Check Property dialog box. Repeat steps 1 and 2.

2. Click **Paste** to paste the statement into your test script.

EXPERIMENT 3

GUI CHECKPOINT FOR SINGLE OBJECT/WINDOW

You can create a GUI checkpoint to check a single object in the application being tested. You can either check the object with its default properties or you can specify which properties to check.

Each standard object class has a set of default checks. For a complete list of standard objects, the properties you can check, and default checks, see "Property Checks and Default Checks".

Creating a GUI Checkpoint using the Default Checks

You can create a GUI checkpoint that performs a default check on the property recommended by WinRunner. For example, if you create a GUI checkpoint that checks a push button, the default check verifies that the push button is enabled.

To create a GUI checkpoint using default checks:

1. Choose **Insert > GUI Checkpoint >for Object/Window**, or click the **GUI Checkpoint for Object/Window** button on the User toolbar. If you are recording in Analog mode, press the CHECK GUI FOR OBJECT/WINDOW soft key in order to avoid extraneous mouse movements. Note that you can press the CHECK GUI FOR OBJECT/WINDOW soft key in Context Sensitive mode as well.

   The WinRunner window is minimized, the mouse pointer becomes a pointing hand, and a help window opens on the screen.
2. Click an object.
3. WinRunner captures the current value of the property of the GUI object being checked and storesit in the test's expected results folder. The WinRunner window is restored and a GUI checkpoint is inserted in the test script as an **obj_check_gui** statement.

Creating a GUI Checkpoint by Specifying which Properties to Check

You can specify which properties to check for an object. For example, if you create a checkpoint that checks a push button, you can choose to verify that it is in focus, instead of enabled.

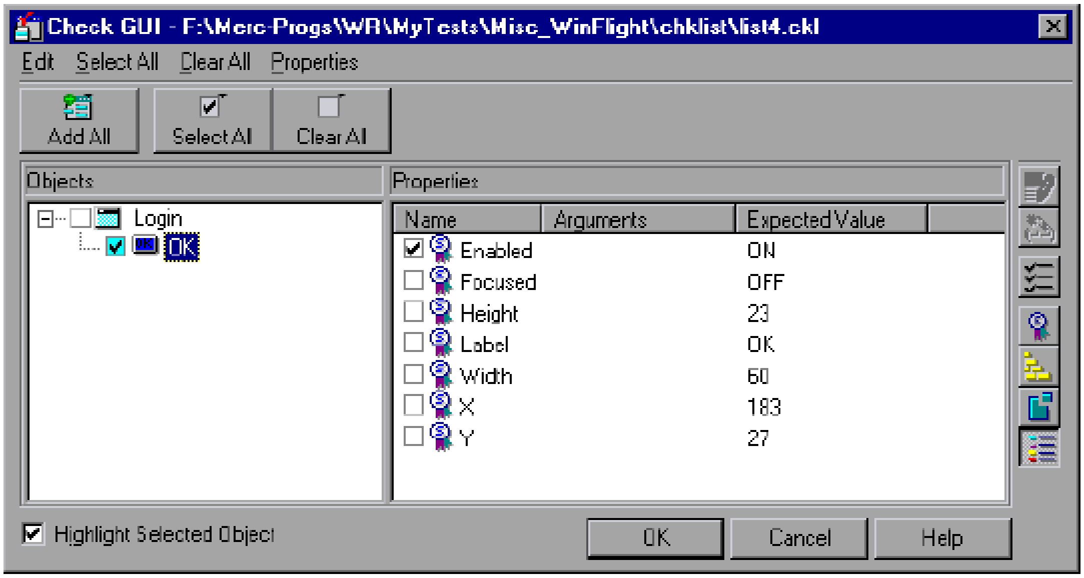
To create a GUI checkpoint by specifying which properties to check:

1. Choose **Insert > GUI Checkpoint >for Object/Window**, or click the **GUI Checkpoint for Object/Window** button on the User toolbar. If you are recording in Analog mode, press the CHECK GUI FOR OBJECT/WINDOW softkey in order to avoid extraneous mouse movements.

Note that you can press the CHECK GUI FOR OBJECT/WINDOW soft key in Context Sensitive mode as well.

The Win Runner window is minimized, the mouse pointer becomes a pointing hand, and a help window opens on the screen.

2. Double-click the object or window. The Check GUI dialog box opens.



Check GUI - F:\Merc-Progs\WR\MyTests\Misc_WinFlight\chklist\list4.ckl

	Name	Arguments	Expected Value
☑	Enabled		ON
□	Focused		OFF
□	Height		23
□	Label		OK
□	Width		60
□	X		183
□	Y		27

☑ Highlight Selected Object

1. Click an object name in the **Objects** pane. The **Properties** pane lists all the properties for the selected object.
2. Select the properties you want to check.
   - To edit the expected value of a property, first select it. Next, either click the **Edit Expected Value** button, or double-click the value in the **Expected Value** column to edit it.
   - To add a check in which you specify arguments, first select the property for which you want to specify arguments. Next, either click the **Specify Arguments** button, or double-click in the **Arguments** column. Note that if an ellipsis (three dots) appears in the Arguments column, then you must specify arguments for a check on this property. (You do not need to specify arguments if a default argument are specified.) When checking standard objects, you only specify arguments for certain properties of edit and static text objects. You also specify arguments for checks on certain properties of nonstandard objects.
   - To change the viewing options for the properties of an object, use the **Show Properties** buttons.
3. Click **OK** to close the Check GUI dialog box.

Win Runner captures the GUI information and stores it in the test's expected results folder. The Win Runner window is restored and a GUI checkpoint is inserted in the test script as an **obj_check_gui** or a **win_check_gui** statement. For more information, see "Understanding GUI Checkpoint Statements".

EXPERIMENT 4

GUI checkpoint for multiple objects

You can use a GUI checkpoint to check two or more objects in a window. For a complete list of standard objects and the properties you can check, see "Property Checks and Default Checks".

To create a GUI checkpoint for two or more objects:

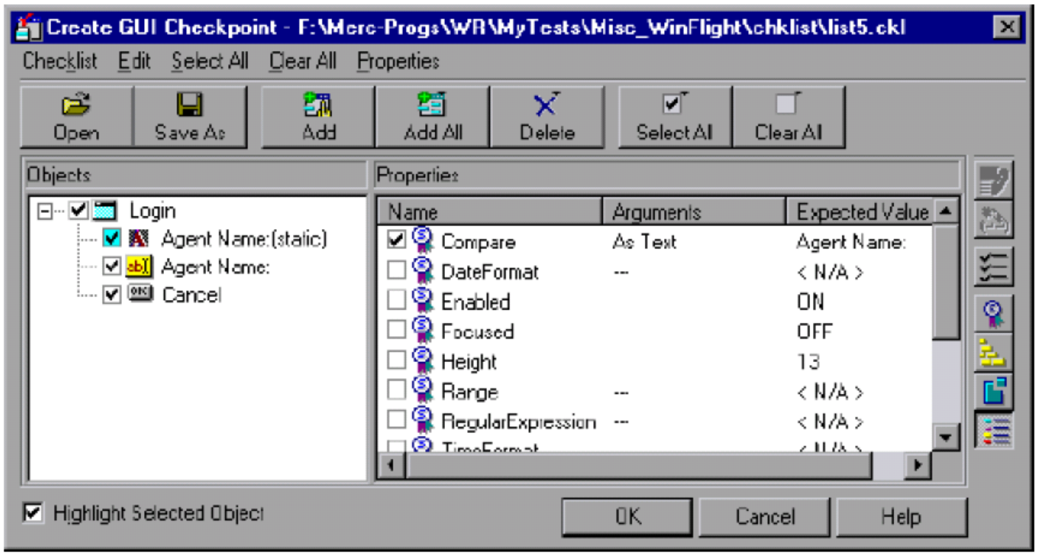
1. Choose **Insert > GUI Checkpoint** > **For Multiple Objects** or click the **GUI Checkpoint for Multiple Objects** button on the User toolbar. If you are recording in Analog mode, press the CHECK GUI FOR MULTIPLE OBJECTS softkey in order to avoid extraneous mouse movements. The Create GUI Checkpoint dialog box opens.

2. Click the **Add** button. The mouse pointer becomes a pointing hand and a help window opens.

3. To add an object, click it once. If you click a window title bar or menu bar, a help window prompts you to check all the objects in the window.

4. The pointing hand remains active. You can continue to choose objects by repeating step 3 above for each object you want to check.

5. Click the right mouse button to stop the selection process and to restore the mouse pointer to its original shape. The Create GUI Checkpoint dialog box reopens.



Create GUI Checkpoint - F:\Merc-Progs\WR\MyTests\Misc_WinFlight\chklist\list5.ckl

☑ Login
- ☑ Agent Name:(static)
- ☑ Agent Name:
- ☑ Cancel

Name	Arguments	Expected Value
☑ Compare	As Text	Agent Name:
□ DateFormat	--	< N/A >
□ Enabled		ON
□ Focused		OFF
□ Height		13
□ Range	--	< N/A >
□ RegularExpression	--	< N/A >
□ TimeFormat		< N/A >

☑ Highlight Selected Object

6. The Objects pane contains the name of the window and objects included in the GUI checkpoint.
To specify which objects to check, click an object name in the **Objects** pane.
The Properties pane lists all the properties of the object. The default properties are selected.

To edit the expected value of a property, first select it. Next, either click the **Edit Expected Value** button, or double-click the value in the **Expected Value** column to edit it.

To add a check in which you specify arguments, first select the property for which you want to specify arguments. Next, either click the **Specify Arguments** button, or double-click in the **Arguments** column. Note that if an ellipsis appears in the Arguments column, then you mustspecify arguments for a check on this property. (You do not need to specify arguments if a default argument is specified.) When checking standard objects, you only specify arguments for certain properties of edit and static text objects.You also specify arguments for checks on certain properties of nonstandard objects.

To change the viewing options for the properties of an object, use the Show Properties buttons.
7. To save the checklist and close the Create GUI Checkpoint dialog box, click **OK**.
WinRunner captures the current property values of the selected GUI objects and stores it in the expected results folder. A **win_check_gui** statement is inserted in the test script.

EXPERIMENT 5

A. Bitmap checkpoint for object/window

You can capture a bitmap of any window or object in your application by pointing to it. The method for capturing objects and for capturing windows is identical. You start by choosing **Insert > Bitmap Checkpoint > For Object/Window**. As you pass the mouse pointer over the windows of your application, objects and windows flash. To capture a window bitmap, you click the window's title bar. To capture an object within a window as a bitmap, you click the object itself.

Note that during recording, when you capture an object in a window that is not the active window, WinRunner automatically generates a **set_window** statement.

To capture a window or object as a bitmap:

● Choose **Insert > Bitmap Checkpoint > For Object/Window** or click the **Bitmap Checkpoint for Object/Window** button on the User toolbar. Alternatively, if you are recording in Analog mode, press the CHECK BITMAP OF OBJECT/WINDOW softkey.

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

● Point to the object or window and click it. WinRunner captures the bitmap and generates a **win_check_bitmap** or **obj_check_bitmap** statement in the script.

The TSL statement generated for a window bitmap has the following syntax:

win_check_bitmap (*object***,** *bitmap***,** *time* **);**

For an object bitmap, the syntax is:

obj_check_bitmap (*object***,** *bitmap***,** *time* **);**

For example, when you click the title bar of the main window of the Flight Reservation application, the resulting statement might be:

win_check_bitmap ("Flight Reservation", "Img2", 1);

However, if you click the Date of Flight box in the same window, the statement might be:

obj_check_bitmap ("Date of Flight:", "Img1", 1);

B. Bitmap checkpoint for screen area

When working in Context Sensitive mode, you can capture a bitmap of a window, object, or of a specified area of a screen. WinRunner inserts a checkpoint in the test script in the form of either a **win_check_bitmap** or **obj_check_bitmap** statement.

To check a bitmap, you start by choosing **Insert > Bitmap Checkpoint**. To capture a window or another GUI object, you click it with the mouse. To capture an area bitmap, you mark the area to be checked using a crosshairs mouse pointer.

Note that when you record a test in Analog mode, you should press the CHECK BITMAP OF WINDOW softkey or the CHECK BITMAP OF SCREEN AREA softkey to create a bitmap checkpoint. This prevents WinRunner from recording extraneous mouse movements. If you are programming a test, you can also use the Analog function **check_window** to check a bitmap.

If the name of a window or object varies each time you run a test, you can define a regular expression in the GUI Map Editor. This instructs WinRunner to ignore all or part of the name. For more information on using regular expressions in the GUI Map Editor, see "Editing the GUI Map."

Your can include your bitmap checkpoint in a loop. If you run your bitmap checkpoint in a loop, the results for each iteration of the checkpoint are displayed in the test results as separate entries. The results of the checkpoint can be viewed in the Test Results window. For more information, see “Analyzing Test Results.”