

WinRunner[®] 2000
Testing for Year 2000

mercury
interactive

Testing for Year 2000

© Copyright 1997 by Mercury Interactive Corporation

All rights reserved. All text and figures included in this publication are the exclusive property of Mercury Interactive Corporation, and may not be copied, reproduced, or used in any way without the express permission in writing of Mercury Interactive. Information in this document is subject to change without notice and does not represent a commitment on the part of Mercury Interactive.

Patents pending

XRunner, WinRunner, and LoadRunner are registered trademarks of Mercury Interactive Corporation. TestDirector, TestSuite, Visual Testing, SMARTest, RapidTest, TSL and Context Sensitive are trademarks of Mercury Interactive Corporation.

This document also contains Registered Trademarks, Trademarks and Service Marks that are owned by their respective companies or organizations. Mercury Interactive Corporation disclaims any responsibility for specifying which marks are owned by which companies or organizations.

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

Mercury Interactive Corporation
470 Potrero Avenue
Sunnyvale, CA 94086
Tel. (408) 523-9900
Fax. (408) 523-9911

WRY2KUG4.05/01

Table of Contents

Chapter 1: Introduction	1
Converting Applications for the Year 2000	1
Testing with WinRunner 2000	2
Assessment Testing Workflow	4
Year 2000 Compliance Testing Workflow	5
Chapter 2: Creating Year 2000 Tests	7
About Creating Year 2000 Tests	7
Setting Date Formats	9
Recording a Year 2000 Test Script	11
Inserting Date Checkpoints	11
Inserting Year 2000 GUI Checkpoints	12
Enhancing Your Year 2000 Test Scripts with TSL	15
Chapter 3: Overriding Date Formats	17
About Overriding Date Formats	17
Overriding Non-Date Fields	18
Overriding Date Formats	19
Overriding Date Formats with TSL	20
Chapter 4: Running Year 2000 Tests	21
About Running Year 2000 Tests	21
Setting the Year 2000 Run Mode	23
Changing Year 2000 Run Mode Settings with TSL	24
Chapter 5: Viewing Year 2000 Test Results	25
About Viewing Test Results	25
Viewing the Results of a Date Checkpoint	25
Viewing the Results of a GUI Checkpoint	28
Index	31

1

Introduction

Welcome to WinRunner 2000, Mercury Interactive's automated Year 2000 testing tool for Graphical User Interface (GUI) applications, and mainframe and AS/400 applications running on 3270 and 5250 protocol terminal emulators.

This guide describes how to test your application for Year 2000 compliance. If you are testing a GUI application, it is recommended that before you begin Year 2000 testing, you first review the *WinRunner User's Guide*. If you are testing a terminal emulator application, it is recommended that you read the *WinRunner User's Guide* and *Testing Terminal Emulator Applications* guide.

This chapter describes:

- Converting applications for the Year 2000
- Testing with WinRunner 2000
- Performing Assessment testing
- Performing Year 2000 Compliance testing

Converting Applications for the Year 2000

In the past, programmers wrote applications using two-character fields to manipulate and store dates (for example '75' represented 1975). Using a two-character date conserved memory and improved application performance at a time when memory and processing power were expensive.

Many of these applications are still in use today, and will continue to be in use into the 21st century. In industries where age calculation is routinely performed, such as banking and insurance, applications using the two-character date format will generate serious errors after December 31, 1999.

For example, suppose in the year 2001 an insurance application attempts to calculate a person's current age by subtracting his birth date from the current date. If the application uses the two-character date format, a negative age will result (Age = 01 - 30 years = -29).

In order to ensure that applications can accurately process date information in the 21st century, programmers must examine millions of lines of code to find date-related functions. Each instance of a two-character date format must be corrected using one of the following methods:

► **Windowing**

Programmers keep the two-character date format, but define thresholds (cut-year points) that will determine when the application recognizes that a date belongs to the 21st century. For example, if 2060 is selected as the threshold, the application recognizes all dates from 0 to 60 as 21st century dates. All dates from 61 to 99 are recognized as 20th century dates.

► **Date Field Expansion**

Programmers expand two-character date formats to four-characters. For example, "98" is expanded to "1998".

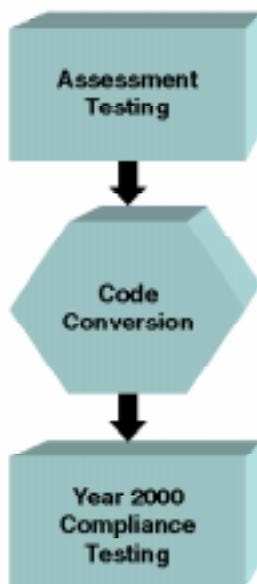
However, programmers cannot simply convert code and put an application back into operation. It is crucial that applications undergo comprehensive testing to detect any date-related defects before the year 2000.

Testing with WinRunner 2000

WinRunner 2000 enables you to create automated tests that check date information in GUI applications and terminal emulator applications. For GUI applications, it records your actions in terms of the GUI objects (such as windows, lists, and buttons) you select. For terminal emulator applications, it records the operations you perform in the context of screens, fields, and PF keys. As you record, WinRunner 2000 lets you insert checkpoints into your test script which check the dates in the application.

To run a test, you simply play back the operations you performed during recording. WinRunner simulates a human user by entering keyboard and mouse input into your application.

WinRunner 2000 assists in two phases of the Year 2000 conversion process: Assessment testing and Year 2000 Compliance testing.



Assessment Testing

Assessment testing helps you identify Year 2000 problems in your application before converting the application. You create a baseline of tests containing checkpoints which check the dates on each screen. To create a test, you record normal business processes performed by your application.

Afterwards, you use *aging* to simulate how the application would process date information after December 31, 1999.

For more information, see “Assessment Testing Workflow” on page 4 in this chapter.

Year 2000 Compliance Testing

Year 2000 compliance testing enables you to identify defects in your application after conversion.

- If programmers used the Windowing conversion method, you simply run the tests again using *aging*.
- If programmers used the Date Field Expansion method, you first run the tests created during Assessment testing, using the new date formats. WinRunner translates all the original dates in the test scripts to the new date format. For example, it could translate all MM/DD/YY dates to MM/DD/YYYY.

After verifying that your tests run successfully using the new format, you then run the tests again using aging to simulate conditions after December 31, 1999.

For more information, see “Year 2000 Compliance Testing Workflow” on page 5 in this chapter.

Assessment Testing Workflow

Assessment testing helps you locate date-related problems in your application before you start the Year 2000 conversion process.

The recommended workflow is as follows:

- 1** Define the date format(s) currently used in your application, for example, DD/MM/YY. For more information refer to Chapter 2, “Creating Year 2000 Tests.”
- 2** Create baseline tests. Record tests on the current version of the application (pre-year 2000 conversion). While recording, insert checkpoints that will check the dates in the application. For more information, refer to Chapter 2, “Creating Year 2000 Tests.”
- 3** Run the tests (in Debug mode) to check that they run smoothly. For more information refer to Chapter 4, “Running Year 2000 Tests.”

If a test incorrectly identifies non-date fields as date fields or reads a date field using the wrong date format, you can override the automatic date recognition on selected fields. For more information, refer to Chapter 3, “Overriding Date Formats.”

- 4** Run the test (in Update mode) to create expected results. For more information refer to Chapter 4, “Running Year 2000 Tests.”

- 5** Run the tests with aging (in Verify mode) to simulate conditions in the 21st century. For more information refer to Chapter 4, “Running Year 2000 Tests.”
- 6** Analyze test results to pinpoint where date problems exist in the application. For more information refer to Chapter 5, “Viewing Year 2000 Test Results.”

Year 2000 Compliance Testing Workflow

Year 2000 compliance testing enables you to identify defects in your application after conversion.

Year 2000 Compliance Testing for Date Field Expansion

If programmers used the Date Field Expansion method to convert an application, the recommended workflow is as follows:

- 1** Define the new date format(s) appearing in the application after conversion, for example, MM/DD/YYYY. For more information refer to Chapter 2, “Creating Year 2000 Tests.”
- 2** Run the tests (in Verify mode) to check that they execute smoothly using the new date format(s). For more information refer to Chapter 4, “Running Year 2000 Tests.”
- 3** Run tests (in Verify mode) with the new date formats *and* aging to simulate conditions in the 21st century. For more information refer to Chapter 4, “Running Year 2000 Tests.”
- 4** Analyze test results to pinpoint where date problems still exist in the application. For more information refer to Chapter 5, “Viewing Year 2000 Test Results.”

Year 2000 Compliance Testing for Windowing

If programmers used the Windowing method to convert an application, the recommended workflow is as follows:

- 1** Run the tests created during the Assessment testing phase with aging to simulate dates in the 21st century. For more information refer to Chapter 4, “Running Year 2000 Tests.”
- 2** Analyze test results to pinpoint where date problems still exist. For more information refer to Chapter 5, “Viewing Year 2000 Test Results.”

2

Creating Year 2000 Tests

You start the Year 2000 testing process by creating a baseline of automated tests, and adding checkpoints to the test scripts to verify dates.

This chapter describes

- ▶ Setting date formats
- ▶ Recording a test script
- ▶ Inserting Date Checkpoints
- ▶ Inserting GUI Checkpoints
- ▶ Enhancing your Year 2000 test scripts with TSL

About Creating Year 2000 Tests

In order to test your application, you use recording to create a baseline of automated tests. As you record, WinRunner 2000 generates a test script in Mercury Interactive's Test Script Language (TSL). You can use programming to further enhance your recorded test script with additional TSL functions and programming elements.

To check date information in your application, you add checkpoints to your test script. When you add a checkpoint, WinRunner looks for dates in the active window or screen, captures the dates, and stores them as expected results. When you run a test, a checkpoint compares the expected date to the actual date displayed in the application. Two types of checkpoints are used in Year 2000 test scripts:

- *GUI checkpoints* capture date information in a GUI application, or in a single date field in a terminal emulator application.
- *Date checkpoints* capture all date information displayed in the active screen of a terminal emulator application.

Note that in order for WinRunner to recognize dates, before you add checkpoints to your application, you must tell WinRunner which date formats are used in your application (such as MMDDYY, DDMMYY or DD/MM/YY).

Sample Test for a GUI Application

The following is a sample WinRunner 2000 test recorded on a GUI application used to reserve airline tickets. The user opens a new ticket order and enters the date of the flight. A GUI checkpoint checks that the date is correct.

```
# Direct input to the main Flight Reservation window.  
set_window ("Flight Reservation", 10);  
  
# Open a new ticket order.  
menu_select_item ("File;New Order");  
  
# Enter the flight date in the "Date of Flight" field.  
edit_set_insert_pos ("Date of Flight:", 0, 0);  
type ("121397");  
  
# Check the date (using a GUI checkpoint).  
obj_check_gui("Date of Flight:", "list1.ckl", "gui1", 1);
```

Sample Test for a Terminal Emulator Application

The following is a sample WinRunner 2000 test recorded on a terminal emulator application. The user presses the Enter key in the first screen of the application. WinRunner waits for the screen to change, and the user types the date 12/13/97 in a field. A date checkpoint checks that the date is correct. The comment (#) lines describe each statement in the script.

```
# Activate the terminal emulator window.  
win_activate("RUMBA - DEMO");
```

```
# Press the Enter key.  
TE_send_key (TE_ENTER);  
  
# Wait for the next screen to refresh.  
TE_wait_sync();  
  
# Direct input to the Logon screen.  
set_window("Logon");  
  
# Type in the current date.  
TE_edit_field("121397");  
  
# Check the date (using a date checkpoint).  
Y2K_check_date ("y2k1");
```

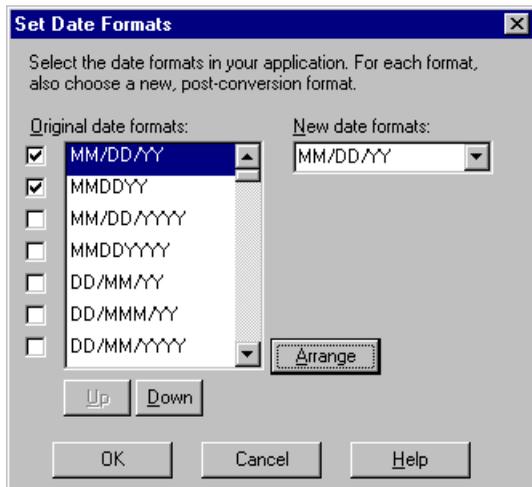
Note: For more information on TE_ functions, see the *Testing Terminal Emulator Applications* guide and the *TSL Online Reference*.

Setting Date Formats

WinRunner 2000 supports a wide range of date formats. Before you begin creating tests, you should specify the date formats currently used in your application. This enables WinRunner to recognize date information when you insert checkpoints into a test script and run tests.

To specify date formats:

- 1** In the Year2000 menu, select Set Date Formats. The Set Date Formats dialog box opens.



- 2** In the Original Date Formats list, click the checkbox next to each date format used in your application.
- 3** Click Arrange to move all selected date formats to the top of the list. You can also use the Move Up and Move Down buttons to rearrange the formats.
- 4** If you are performing Year 2000 Compliance testing, select the post-conversion date format used in your application from the New Date Format list.

Note that if the windowing conversion method was used, the original date format and the new date format should be identical.

If the date format used in your application does not appear in the list, custom date formats can be defined. Contact Mercury Interactive Customer Support for more information.

- 5** Click OK.

Recording a Year 2000 Test Script

You use the Context Sensitive recording mode to create a Year 2000 test script.

To record a Year 2000 test script:

- 1** Select Create > Record - Context Sensitive to start recording, or press the Record icon.
- 2** Use the mouse and keyboard to perform operations in your application.
- 3** While recording, insert Date checkpoints and GUI checkpoints into the test script to check dates. For more information see “Inserting Date Checkpoints” on page 11 and “Inserting Year 2000 GUI Checkpoints” on page 12.
- 4** Select Create > Stop Recording to stop the test, or press the Stop icon.
- 5** Enhance the recorded test script with additional Year 2000 functions (optional).
- 6** Select File > Save to save the test script.

Inserting Date Checkpoints

Date checkpoints capture all date information in a screen of a terminal emulator application. When you run the test, WinRunner 2000 compares the original dates captured when you created the test to the actual dates in the application.

Note that if you want to check date information in a single date field in a terminal emulator application, use a GUI Checkpoint. For more information. See “Inserting Year 2000 GUI Checkpoints” on page 12.

Inserting Date Checkpoints Manually

You can manually add a date checkpoint to your test script while recording. To do so, simply press the CHECK DATE softkey (Ctrl_L F2 for AS/400 applications and Ctrl_L PgDn for mainframe applications).

WinRunner inserts the following statement into the test script:

```
Y2K_check_date(name);
```

The *name* parameter is the name of the date checkpoint. For example in the statement:

```
Y2K_check_date("y2k3");
```

y2k3 indicates the third date checkpoint in the test script.

Note: You can change the softkey configuration of a date checkpoint using the Softkey Configuration utility. See the *WinRunner User's Guide* for more information.

Inserting Date Checkpoints Automatically

You can set WinRunner to automatically insert a date checkpoint into the test script each time you change screens in your terminal emulator application. To do so, add the following function at the top of a test script:

```
Y2K_set_auto_date_verify(ON/OFF);
```

For more information on `Y2K_set_auto_date_verify`, see the *TSL Online Reference*.

Inserting Year 2000 GUI Checkpoints

GUI checkpoints verify information about GUI objects and fields in your application. If you are testing a GUI application, you can add a GUI checkpoint that checks date information in an edit box or a static text object. If you are testing a terminal emulator application, you can add a GUI checkpoint to your test script that checks date information in a single field.

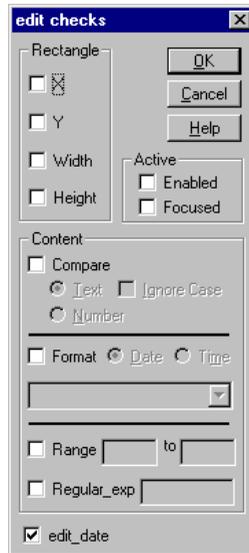
To check the date in a GUI object:

- 1** Select Check GUI > Object/Window from the Create menu.

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

- 2 To check only the date, click on the edit box or static text object once.

To check additional information about the object, double-click on the field to bring up the appropriate dialog box.



Select checks and click OK to close the dialog box.

WinRunner captures the field information and stores it in the test's expected results directory. The WinRunner window is restored and a GUI checkpoint is inserted in the test script.

For an edit box or static text object, a GUI checkpoint has the following syntax:

obj_check_gui(object, checklist, GUI_file, time);

For example, if you click on a field in a terminal emulator application, the resulting statement might be:

```
obj_check_gui("Date of Flight:", "list1.ckl", "gui1", 1);
```

For more information on **obj_check_gui**, see the *TSL Online Reference*. For additional information on GUI checkpoints, see the *WinRunner User's Guide*.

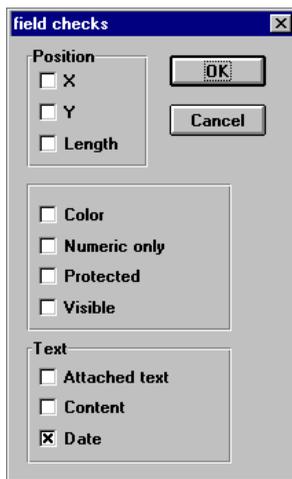
To check the date in a single field in a terminal emulator application:

- 1 Select Check GUI > Object/Window from the Create menu.

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

- 2 To check only the date, click on the field once.

To check additional information about the object, double-click on the field to bring up the Field Checks dialog box.



Select checks and click OK to close the dialog box.

WinRunner captures the field information and stores it in the test's expected results directory. The WinRunner window is restored and a GUI checkpoint is inserted in the test script.

For a field, a GUI checkpoint has the following syntax:

obj_check_gui(*date_field*, *checklist*, *GUI_file*, *time*);

For example, if you click on a field in a terminal emulator application, the resulting statement might be:

```
obj_check_gui("Date:", "list1.ckl", "gui1", 1);
```

For more information on **obj_check_gui**, see the *TSL Online Reference*. For additional information on GUI checkpoints, see the *WinRunner User's Guide*.

Enhancing Your Year 2000 Test Scripts with TSL

You can enhance your recorded Year 2000 test scripts by adding the following TSL functions.

- ▶ The **Y2K_align_day** function enables you to tell WinRunner to age dates to a business day or to the same day of the week.

```
Y2K_align_day ( mode );
```

- ▶ The **Y2K_calc_days_in_field** function calculates the number of days between two date fields.

```
Y2K_calc_days_in_field ( field_name1, field_name2);
```

- ▶ The **Y2K_calc_days_in_string** function calculates the number of days between two numeric strings.

```
Y2K_calc_days_in_string ( string1, string2);
```

- ▶ The **Y2K_field_to_Julian** translates the contents of a date field to a Julian number.

```
Y2K_field_to_Julian ( date_field );
```

- ▶ The **Y2K_is_leap_year** determines whether a year is a leap year.

```
Y2K_is_leap_years ( year );
```

- ▶ The **Y2K_string_to_Julian** translates the contents of a numeric string to a Julian number.

```
Y2K_string_to_Julian ( string );
```

For more information on Year 2000 TSL functions, see the *TSL Online Reference*.

3

Overriding Date Formats

You can avoid errors when running tests by overriding how WinRunner 2000 identifies selected fields.

This chapter describes:

- ▶ Overriding non-date fields
- ▶ Overriding date formats
- ▶ Overriding date formats with TSL

About Overriding Date Formats

After you create a baseline of tests, you should debug your test scripts to check that WinRunner 2000 correctly identifies dates in your application. As you debug your tests, you may find that WinRunner:

- ▶ identifies a non-date field as date field, if the information in the field looks like a date.

For example, suppose an application contains catalog numbers such as 123172. If you defined the format MMDDYY as one of your application's date formats, WinRunner identifies the catalog number as a date field.

- ▶ ignores an irregular date field or reads the date field using the wrong date format.

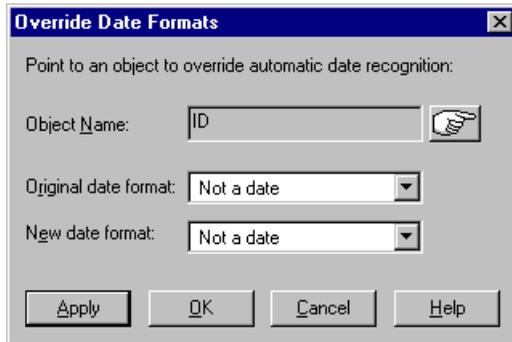
You can override how WinRunner identifies date fields and non-date fields using the Override Date Formats dialog box.

Overriding Non-Date Fields

You can tell WinRunner to ignore objects that look like date fields, but contain non-date information.

To override how WinRunner recognizes non-date fields:

- 1 Select Year2000 > Override Date Formats. The Override Date Formats dialog box opens.



- 2 Click on the pointing hand button and then click on the object in your application.

WinRunner displays the name of the selected object in the Object Name field.

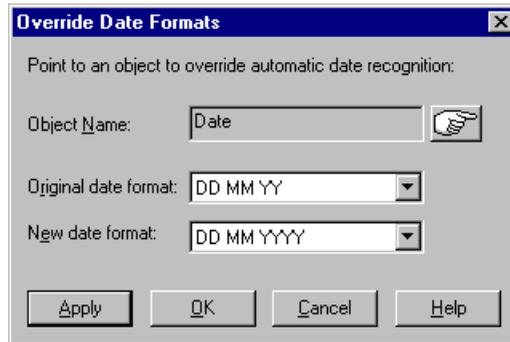
- 3 In the Original Date Format list, select "Not a date".
- 4 In the New Date Format list, select "Not a date".
- 5 Click Apply. You can continue to use the Override Date Formats dialog box to override additional objects.
- 6 Click OK to close the dialog box.

Overriding Date Formats

If WinRunner identifies a date field with the wrong date format, you can override the incorrect format and assign the field a new format.

To override a date format:

- 1 Select Year2000 > Override Date Formats. The Override Date Formats dialog box opens.



- 2 Click on the pointing hand button and then click on the object in your application.

WinRunner displays the name of the selected object in the Object Name field.

- 3 In the Original Date Format list, select the object's original date format.
- 4 In the New Date Format list, select the object's new date format after the application is converted.
- 5 Click Apply. You can continue to use the Override Date Formats dialog box to override additional fields.
- 6 Click OK to close the dialog box.

Overriding Date Formats with TSL

You can override the date format for a date field by using the following TSL function in a test script:

Y2K_change_original_new_formats(*field name, original format, new format*);

For more information, see the *TSL Online Reference*.

4

Running Year 2000 Tests

Once you have developed a baseline of tests, you run the tests to check how your application calculates date information.

This chapter describes:

- Setting the Year 2000 run mode
- Changing run mode settings with TSL

About Running Year 2000 Tests

When you run a Year 2000 test script, WinRunner interprets the test script, line by line, and operates your application as if a person was at the controls. At each checkpoint in the test script, it compares the expected dates with the actual dates appearing in your application.

To run a Year 2000 test script, you choose a general run mode and Year 2000 run mode. Three general run modes are available:

- *Verify mode*, to check your application
- *Debug mode*, to debug your test script
- *Update mode*, to update the expected results

To set the Year 2000 run mode, you also select:

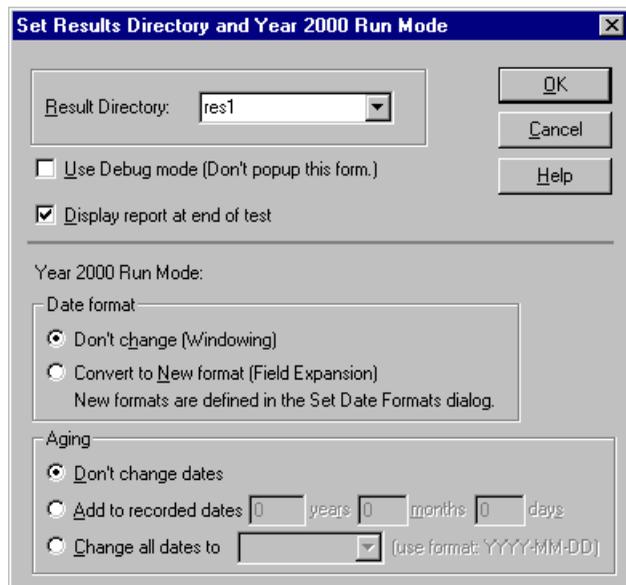
- a *Date format*, to determine whether WinRunner uses the original date formats or the new date formats
- *Aging*, to simulate a future date

To run a Year 2000 test:

- 1** Set the default Year 2000 run mode in the Year 2000 Run Mode dialog box. (Year2000 > Run Mode).
- 2** Open the test if it is not already open.
- 3** Select a general run mode (Verify, Debug, or Update) from the dropdown list of modes on the toolbar.
- 4** Choose a Run command from the Run menu (Run from Top, Run from Arrow, Run Minimized > From Top, or Run Minimized > From Arrow). For more information on Run commands, see the *WinRunner User's Guide*.

If you selected *Debug mode* or *Update mode*, the test script immediately starts to run using the default Year 2000 run mode settings.

If you selected *Verify mode*, the Set Results Directory and Year 2000 Run Mode dialog box opens.



Assign a name to the test run. Use the default name appearing in the Results Directory field, or type in a new name. If you would like to override the

default Year 2000 run mode settings, select new settings that will apply only to the current test run.

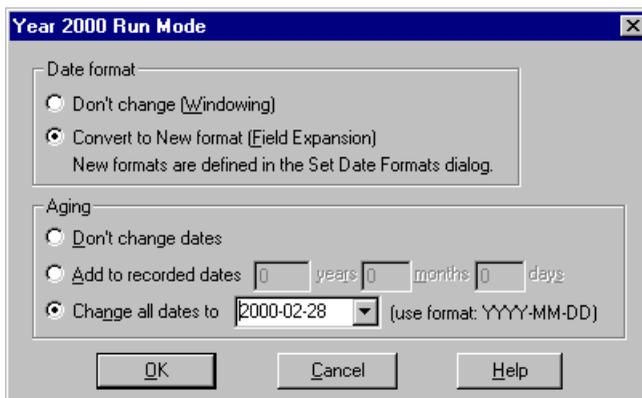
Click OK to close the dialog box and run the test.

Setting the Year 2000 Run Mode

You can set a default Year 2000 run mode that will apply to all your tests.

To set a default Year 2000 run mode:

- 1 Select Run mode from the Year2000 menu. The Year 2000 Run Mode dialog box opens.



- 2 Select a date format.

Select “Don’t Change Dates” if you are performing Assessment testing, or Year 2000 Compliance testing on an application that was converted using the Windowing method.

Select “Convert to New Format” if you are performing Year 2000 Compliance testing on an application that was converted using the Date Field Expansion method.

- 3 Set aging instructions.

Select “Don’t Change Dates” if you are performing Assessment testing.

Select “Add to Recorded Dates” or “Change All Dates to” if you want to simulate a future date during either Assessment testing or Year 2000 Compliance testing.

- 4 Click OK.

Note: You can override the default Year 2000 run mode by changing the default setting in the “Set Run Directory and Year 2000 Run Mode” dialog box. This dialog box opens when you run a test in Verify mode.

Changing Year 2000 Run Mode Settings with TSL

You can set the Year 2000 run mode for a test script using the following TSL functions.

- The `Y2K_set_aging` function ages the test script. It has the following syntax:

```
Y2K_set_aging ( format, type, days, months, years );
```

- The `Y2K_set_replay_mode` function sets the Year 2000 run mode. It has the following syntax:

```
Y2K_set_replay_mode ( mode );
```

- The `Y2K_set_year_threshold` function sets the year threshold (cut-year point). If the threshold is 60, all years from 61 to 99 are recognized as 20th century dates and all dates from 0 to 60 are recognized as 21st century dates. This function has the following syntax.

```
Y2K_set_year_threshold ( number );
```

For more information on Year 2000 TSL functions, see the *TSL Online Reference*.

5

Viewing Year 2000 Test Results

After you execute a Year 2000 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:

- ▶ Viewing the results of a date checkpoint
- ▶ Viewing results of a GUI checkpoint

About Viewing Test Results

When a test run is completed, you can view detailed test results in the WinRunner Reports window. The report contains a description of the major events that occurred during the test run, such as date checkpoints and GUI checkpoints. It also includes tables and pictures to help you quickly analyze the test results.

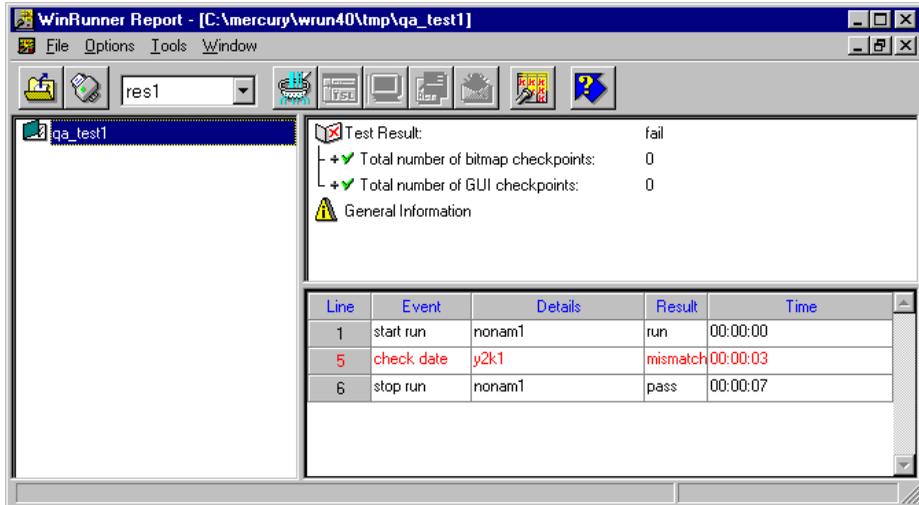
For more information on the WinRunner Reports window, see the *WinRunner User's Guide*.

Viewing the Results of a Date Checkpoint

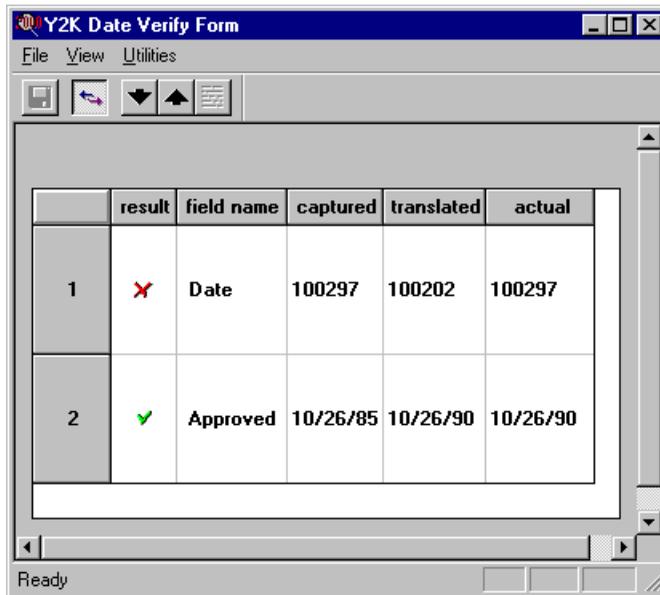
Date checkpoints capture all date information in a screen of a terminal emulator application. When you run a Year 2000 test, WinRunner compares expected dates with the actual dates in the application. After you run a test, you can view the date checkpoint results in a table.

To view the results of a date checkpoint:

- 1 Select Report from the Tools menu. In the test log, look for “check date” entries in the Event column. Passed date checkpoints appear in green; failed date checkpoints appear in red.



- 2 Double-click on a “check date” entry in the test log, or select the entry and press the Display icon. The Y2K Date Verify Form opens:



For each date, the Y2K Date Verify table displays the following information:

result: The result of the check on the field (pass or fail).

field name: The name of the date field checked.

captured: The original date captured.

translated: The date after a new date format and/or aging was applied.

actual: The actual date appearing in the application.

- 3 To view the screen and field containing the captured date, select a check and click the Main Frame View icon. The Terminal Display dialog box opens:



The Source Format screen highlights the date in its original format. The Target Format Screen highlights the date in its new format.

Click OK to close the dialog box.

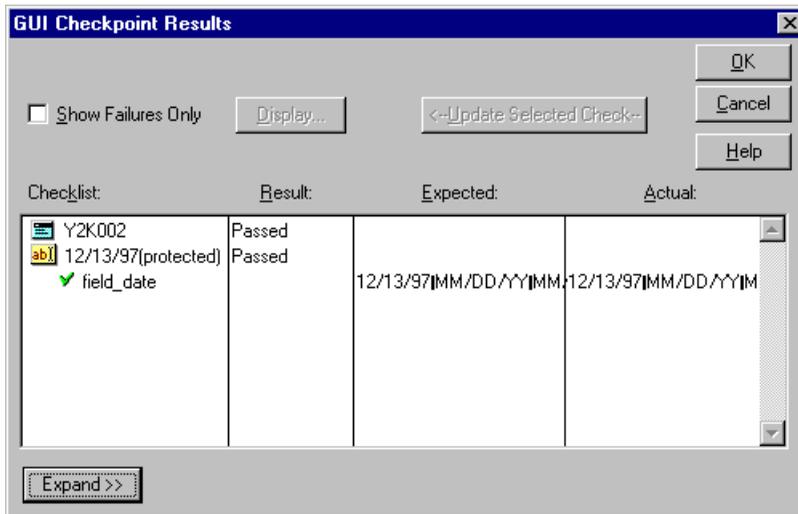
- 4 Select File > Exit to close the Y2K Date Verify Form dialog box.

Viewing the Results of a GUI Checkpoint

GUI checkpoints enable you to check a single field in a screen of a terminal emulator application, or an object in a GUI application. When you run a Year 2000 test, WinRunner compares the expected date with the actual date in the application. To view the results of a GUI checkpoint:

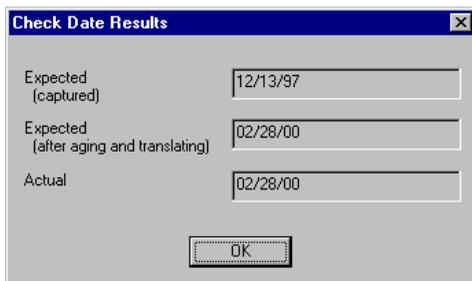
- 1 Open the Report window. In the test log, look for “end GUI checkpoint” entries in the Event column. Passed date checkpoints appear in green; failed date checkpoints appear in red.
- 2 Double-click on a “end GUI checkpoint” entry in the test log, or select the entry and press the Display icon. The GUI Checkpoint Results dialog box opens.

The GUI Checkpoint Results dialog box lists the results of the selected GUI checkpoint in the table.



The table lists the field check and the type of checks performed. Each check is listed as either passed or failed and the expected and verification results are shown.

- 3 To display only failed checks, select the Show Failures Only checkbox.
- 4 To view detailed information about a check on a date, double-click on the check. The Check Date Results dialog box opens.



The Check Date Results dialog box lists the original expected date, the expected date after aging and translation, and the actual date appearing in the object.

Click OK to close the dialog box and to return to the GUI Checkpoints Results dialog box.

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

Index

A

Aging

- defined 4
- Year 2000 run mode 21

Assessment testing

- defined 3
- workflow 4

C

Check Date Results dialog box 29

Check Date softkey 11

Creating tests 7–15

D

Date checkpoints

- inserting automatically 12
- inserting manually 11
- viewing results 25

Date Field Expansion 2

Date formats

- overriding 17–20
- setting 9

Debug mode 21

F

Field checks dialog box 13, 14

G

GUI checkpoints

creating 12

viewing results 28

I

Introduction 1–6

O

obj_check_gui function 13, 14

Override Date Formats command 18

Override Date Formats dialog box 18

Overriding date formats 17–20

R

Recording tests 11

Results, viewing 25–30

- date checkpoints 25

- GUI checkpoints 28

Run Mode command 23

Running tests 21–24

S

Set Date Formats command 10

T

Terminal Display dialog box 28

U

Update mode 21

V

Verify mode 21

W

Windowing 2

Workflow

 assessment testing 4

 Year 2000 Compliance testing 6

Y

Y2K Date Verify Form 27

Y2K_align_day function 15

Y2K_calc_days_in_field function 15

Y2K_calc_days_in_string function 15

Y2K_change_original_new_formats function 20

Y2K_check_date function 12

Y2K_field_to_Julian function 15

Y2K_is_leap_year function 15

Y2K_set_aging function 24

Y2K_set_auto_date_verify 12

Y2K_set_replay_mode function 24

Y2K_set_year_threshold function 24

Y2K_string_to_Julian function 15

Year 2000 Compliance testing

 defined 3

 workflow for Date Field Expansion 5

 workflow for Windowing 6

Year 2000 run mode 21

 aging 21

 date format 21

 setting default 23

Year 2000 Run Mode dialog box 23



Mercury Interactive Corporation
470 Potrero Avenue
Sunnyvale, CA 94086
Tel. (408) 523-9900
Fax. (408) 523-9911