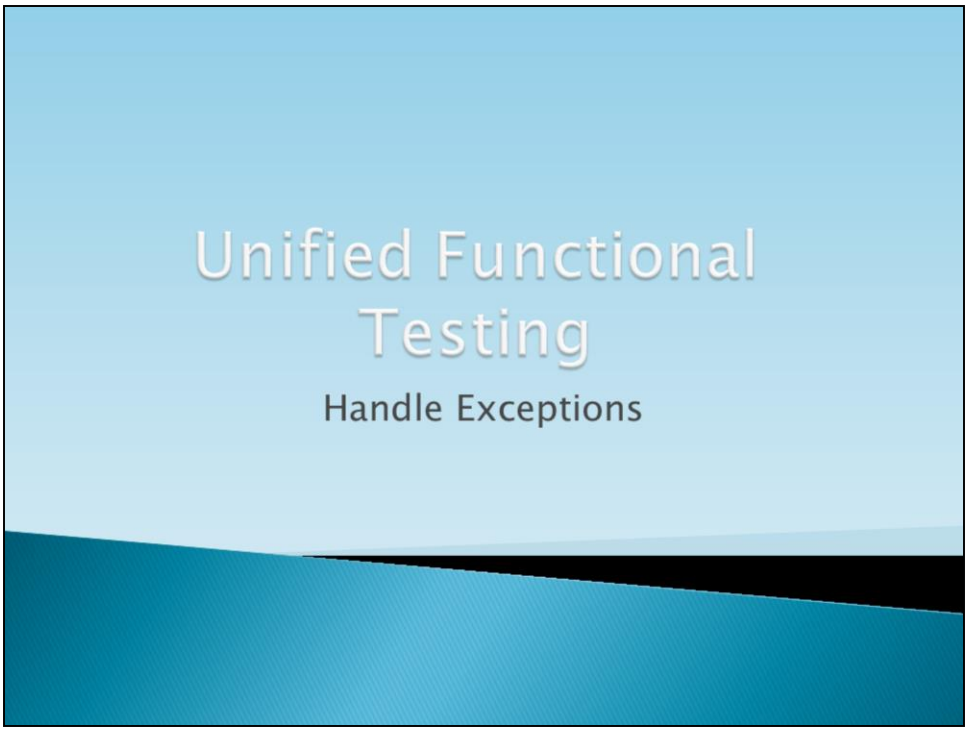

Unified Functional Testing
Handle Exceptions

Lesson Objectives

By the end of this lesson you will be able to:

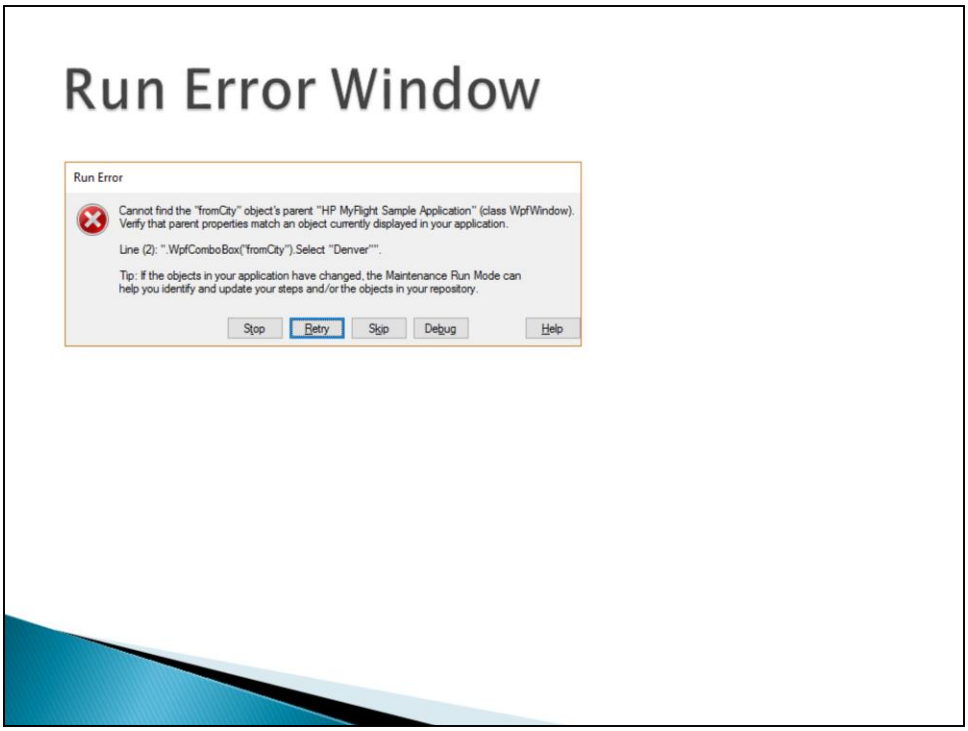
- Identify exceptions in a test.
- Create a recovery scenario.
- Associate a recovery scenario with a test.
- Handle exceptions programmatically.
- Set an optional step in a test

Topics

1. What is an exception
2. Recovery manager
3. Exit statements
4. VBScript error object

Exception – Definition

- Unexpected error and events that occur during an execution and may distort the test execution and may give invalid test results.
- An exception could occur in the AUT or an exception could occur because sources are external to the AUT. For Example , an exception occurs if notification messages, such as you have mail or Printer out of paper, appear during a test run.



When the UFT encounters an exception within a run session, it displays a RUN ERROR window. The RUN ERROR window provides the following options to handle the exceptions:

• **STOP:** Stops the run session. The test results display that the script was not completed.

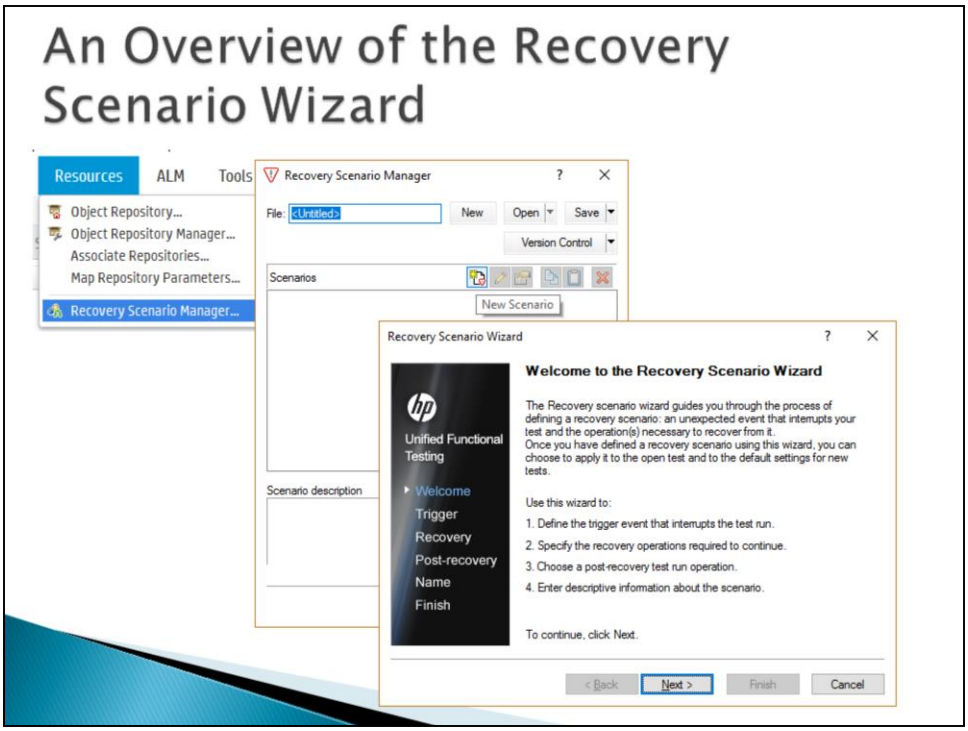
• **RETRY:** Attempts to perform the step again. If the step succeeds, the test run continues.

• **SKIP:** Skips the step that caused the exception and continues the test run from the next step.

• **DEBUG:** Pauses the test run, enabling you to review variable values or step through the script one line at a time.

Advantage of Exception Handling

- Exception handling provides the following advantages:
  - It enables a script to handle exceptions and ensures that the test runs to completion.
  - It enables the testing of positive and negative test data.
- UFT handles exceptions through:
  - Programmatically
  - RECOVERY SCENARIO WIZARD.



To successfully complete a test run, you need to identify the exceptions that can occur during a test run and take appropriate action to handle the exceptions.

You can create recovery scenarios in UFT to handle exceptions that occur during a test run.

To open RECOVERY SCENARIO WIZARD:

1. From the UFT menu bar, select **RESOURCES→RECOVERY SCENARIO MANAGER.** The **RECOVERY SCENARIO MANAGER** dialog box appears.

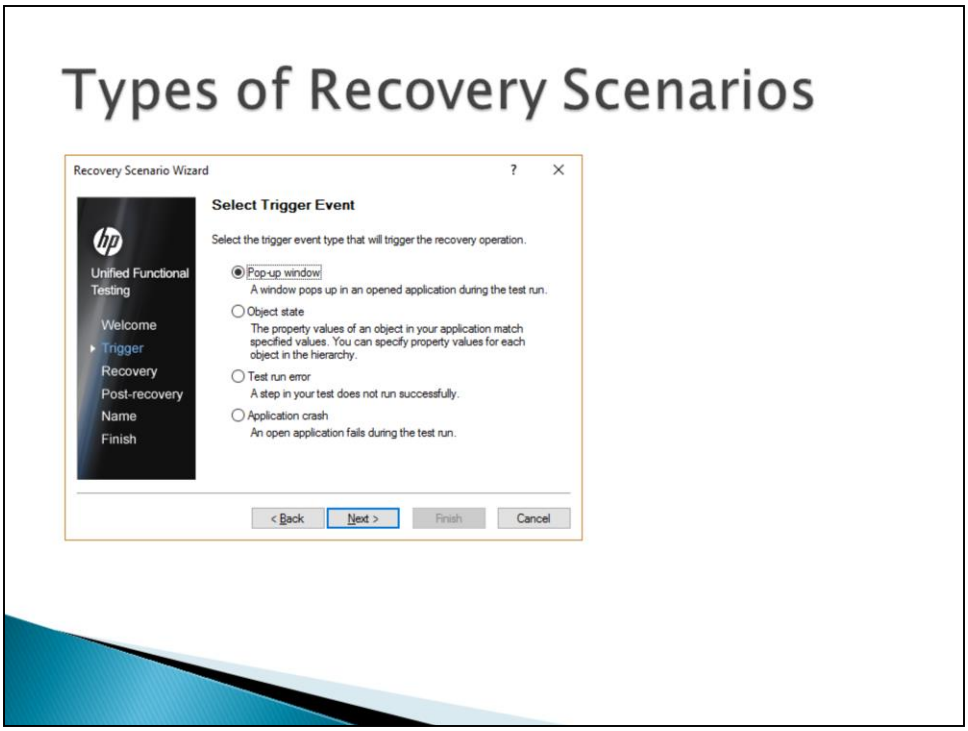
2. In the SCENARIOS section, click the **NEW SCENARIO** button to display **RECOVERY**

SCENARIO WIZARD.

A recovery scenario consists of the following components:

1. **TRIGGER**: Specifies the exception that may occur during a run session. For example, the test may be interrupted by an error pop-up window.
2. **RECOVERY**: Instructs UFT how to handle the exception. For example, you can instruct UFT to click a button on the error pop-up window to close the window.
3. **POST-RECOVERY**: Instructs UFT how to proceed after the recovery operations are performed. For example, after an exception is handled, you may want UFT to restart a test from the beginning.

After you create a recovery scenario, you can associate it with one or more tests.



After opening **RECOVERY SCENARIO WIZARD ,**Click **NEXT.** The **SELECT TRIGGER EVENT** page appears.

The types of recovery scenarios are defined by the type of triggers. The following are

the trigger events in RECOVERY SCENARIO WIZARD:

• **POP-UP WINDOW:** UFT detects a pop-up window and identifies it

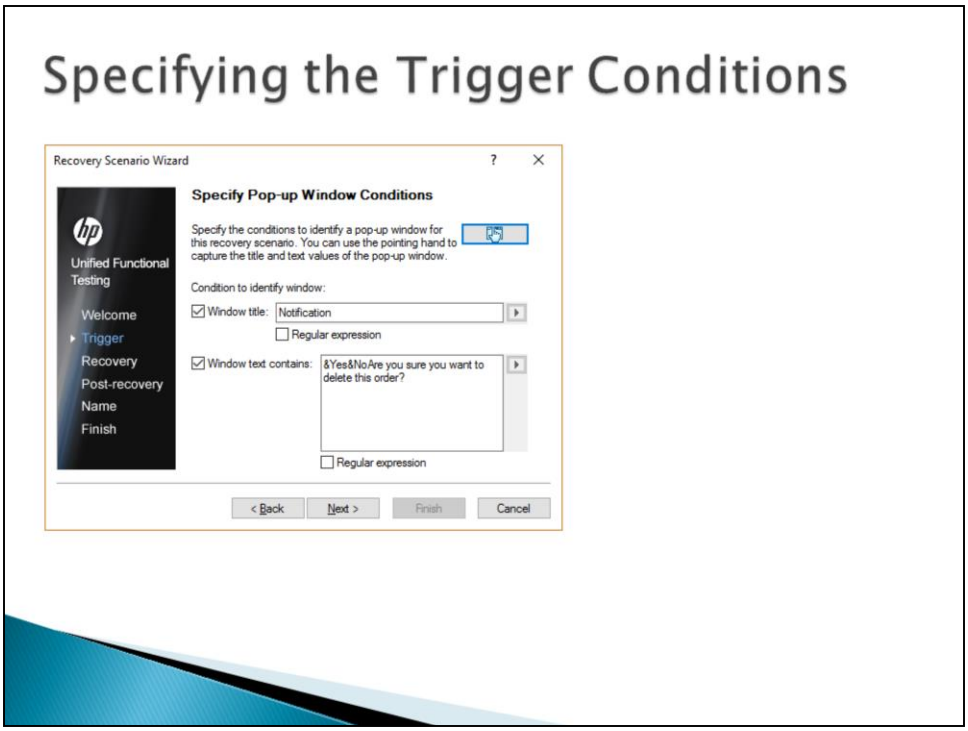
according to the window title and textual content.

• **OBJECT STATE:** UFT detects a specific test object state and identifies it according to the property values of the object and all the objects in the hierarchy.

• **TEST RUN ERROR:** UFT detects an exception and identifies it by a

failed return value from a method.

• **APPLICATION CRASH:** UFT detects an application crash and identifies it according to a predefined list of applications.

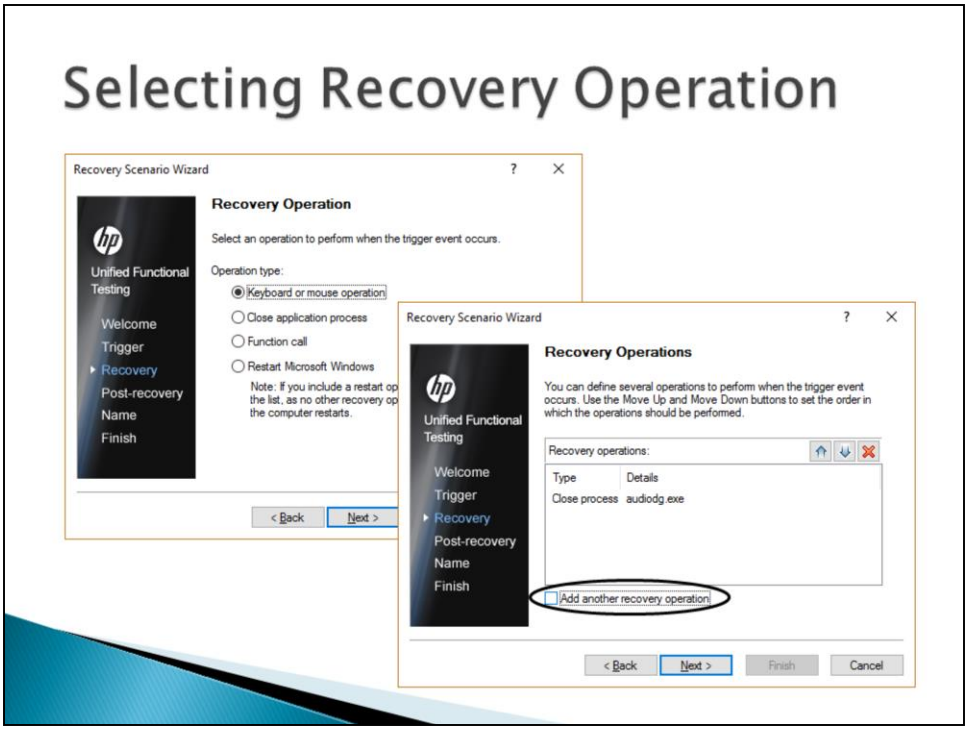


After you select a trigger event type, you specify the conditions that can help UFT identify the correct event during a test run.

To specify the conditions for the POP-UP window trigger event type:

1. On the **SPECIFY POP-UP WINDOW CONDITIONS** page, click the pointing hand button, and click the pop-up error message box that you want the recovery scenario to handle. The **WINDOW TITLE** and the **WINDOW TEXT CONTAINS** fields display the conditions to identify the pop-up window.

2. Click **NEXT**. The **RECOVERY OPERATION** page appears.



After you specify the trigger event type and the trigger event conditions in **RECOVERY SCENARIO WIZARD**, you specify the recovery operation(s) required to handle the trigger

event.

To identify the recovery operation:

1. On the **RECOVERY OPERATION** page of **RECOVERY SCENARIO WIZARD**, select any of the following recovery operations :

– **KEYBOARD OR MOUSE OPERATION**: Performs a key press or a mouse click recovery operation.

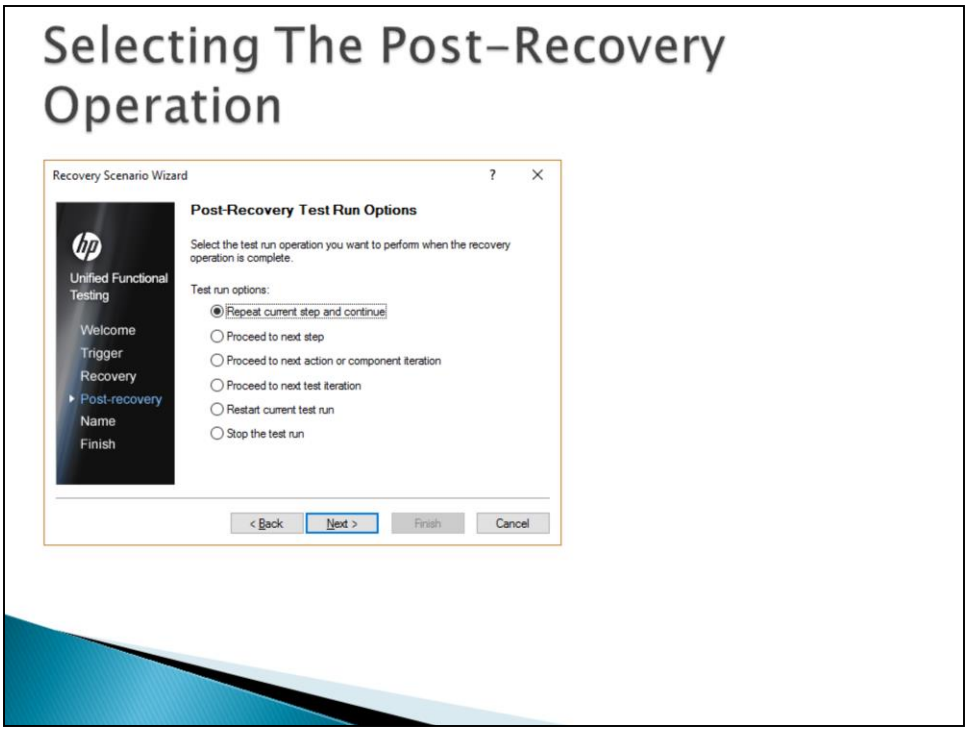
– **CLOSE APPLICATION PROCESS**: Ends the specified application process.

– **FUNCTION CALL**: Calls a VBScript function.

– **RESTART MICROSOFT WINDOWS**: Restarts the Windows operating system.

2. Click **NEXT** to select a recovery operation, such as **CLICK BUTTON WITH LABEL**.

3. Click **NEXT**. The **POST-RECOVERY TEST RUN OPTIONS** page appears.



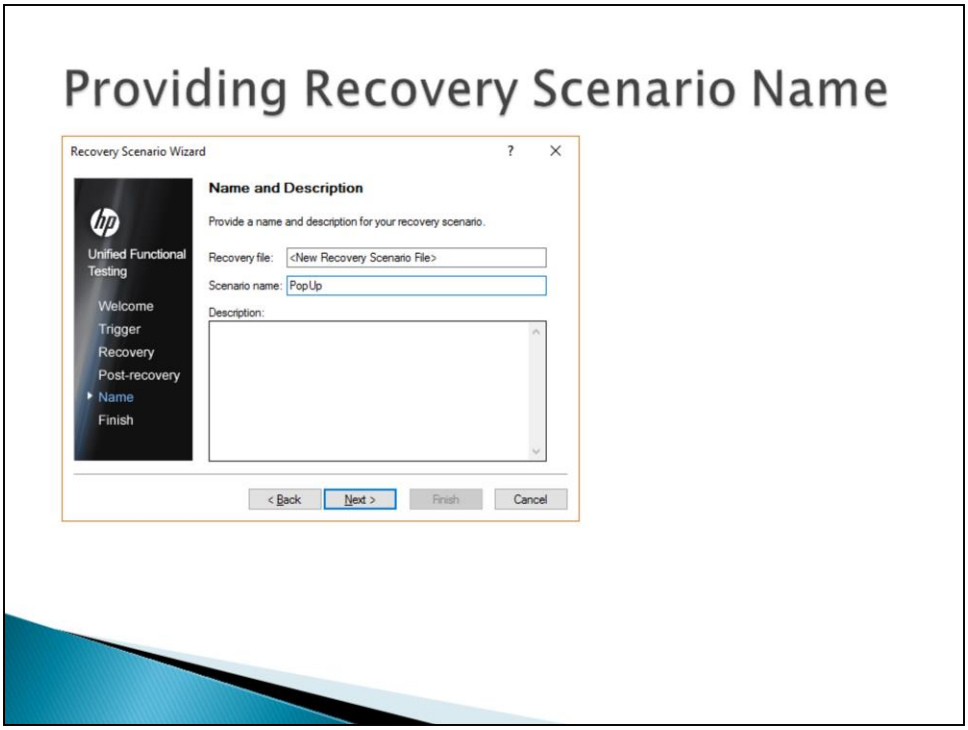
After you specify the recovery operation, identify the operation that you want to perform

when the recovery operation is complete.

To select a post-recovery option:

1. On the **POST-RECOVERY TEST RUN OPTIONS** page, select a post-recovery option.

2. Click **NEXT.** The **NAME AND DESCRIPTION** page appears.



After you identify a trigger event and select the recovery and post-recovery operations

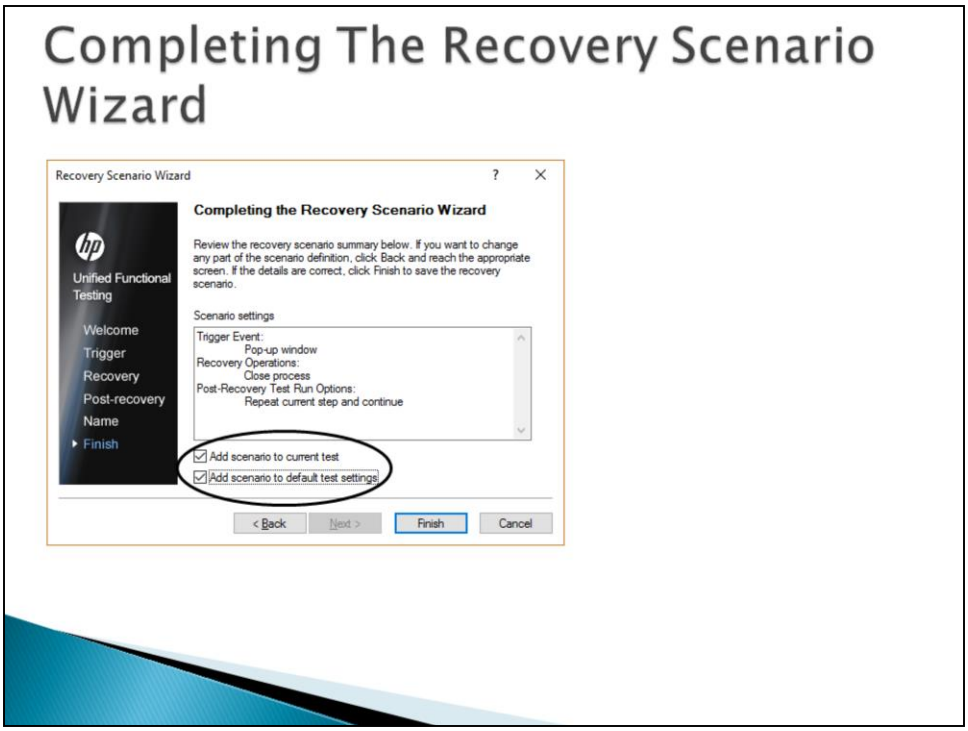
for a recovery scenario, you use the NAME AND DESCRIPTION page to provide a name

and description for the recovery scenario.

To name a recovery scenario:

1. On the NAME AND DESCRIPTION page, specify a name and description for the recovery scenario in the **SCENARIO NAME** and **DESCRIPTION** fields, respectively.

2. Click **NEXT.** The **COMPLETING THE RECOVERY SCENARIO WIZARD** page appears.



The **COMPLETING THE RECOVERY SCENARIO WIZARD** page displays the details of the recovery scenario in the SCENARIO SETTINGS section.

On the **COMPLETING THE RECOVERY SCENARIO WIZARD** page:

1. To associate the recovery scenario with the current test, check the **ADD SCENARIO**

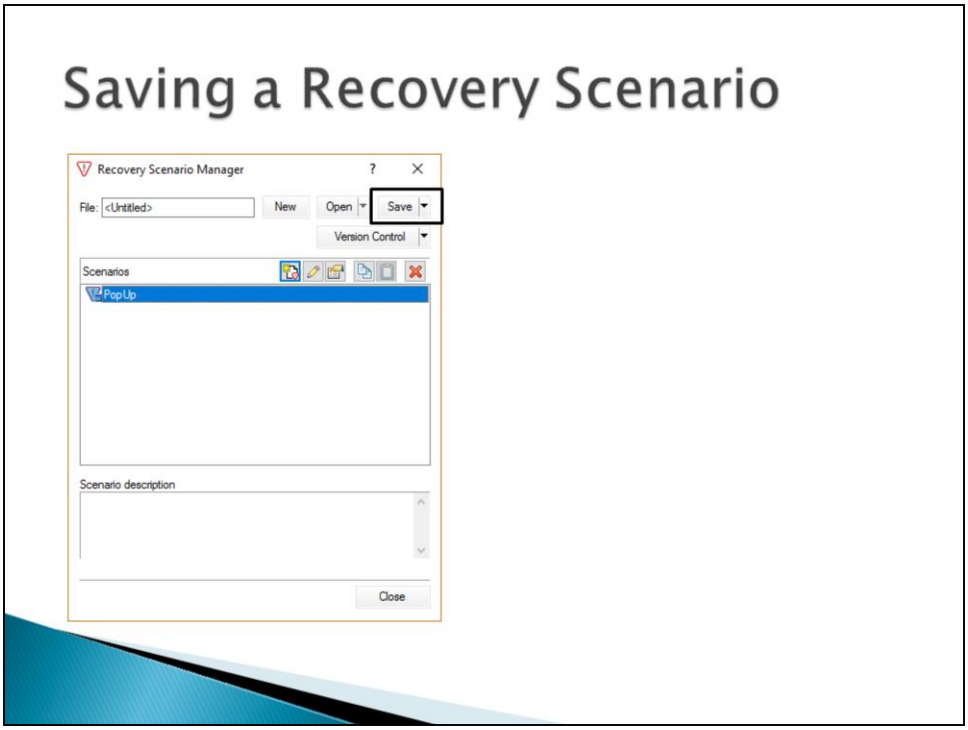
TO CURRENT TEST check box.

2. To associate the recovery scenario with all tests that you create in UFT, check

the **ADD SCENARIO TO DEFAULT TEST SETTINGS** check box.

3. To save the recovery scenario, click **FINISH.** The **RECOVERY SCENARIO MANAGER**

dialog box appears.



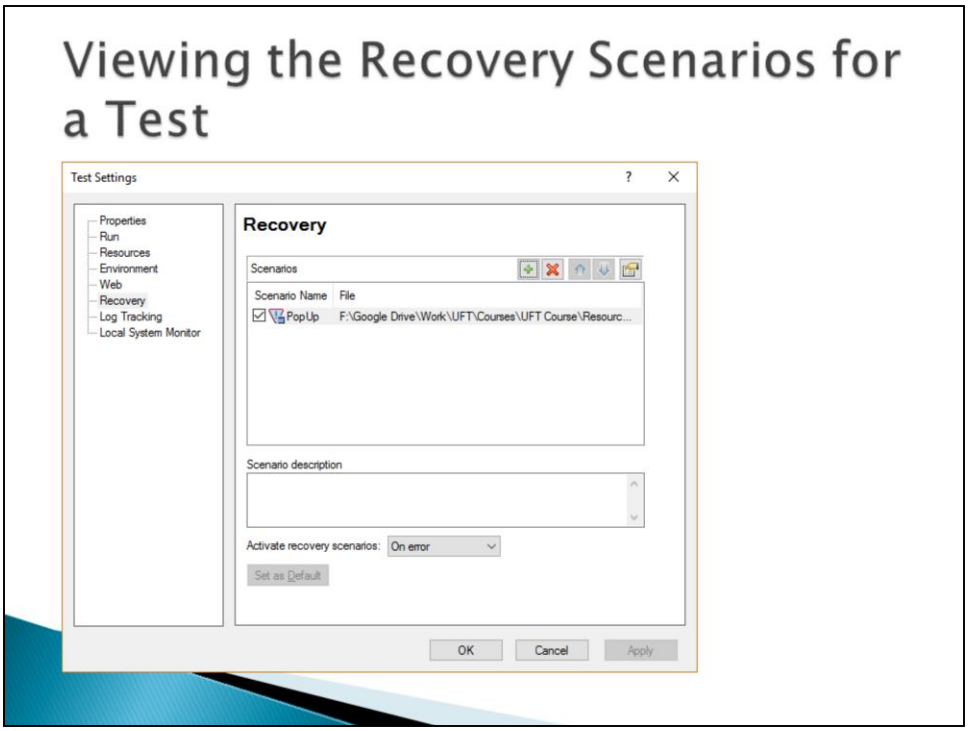
After you create a recovery scenario, save the recovery scenario to a file.

To save a recovery scenario to a file:

1. In **RECOVERY SCENARIO MANAGER**, click **SAVE.**

2. In the **SAVE ATTACHMENT** dialog box, specify a name for the recovery scenario file,

and click **SAVE**. The recovery scenario is saved to the recovery scenario file.

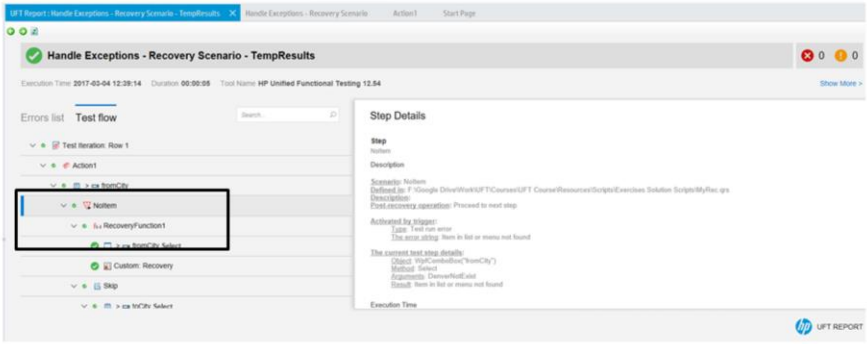


The **RECOVERY** tab of the **TEST SETTINGS** dialog box displays a list of all recovery scenarios associated with the current test. You can associate additional recovery scenarios with a test, remove recovery scenarios from the test, change the order in which they are applied to a run session, and view a read-only summary of each recovery scenario in the **TEST SETTINGS** dialog box.

To add an existing recovery scenario to a test:

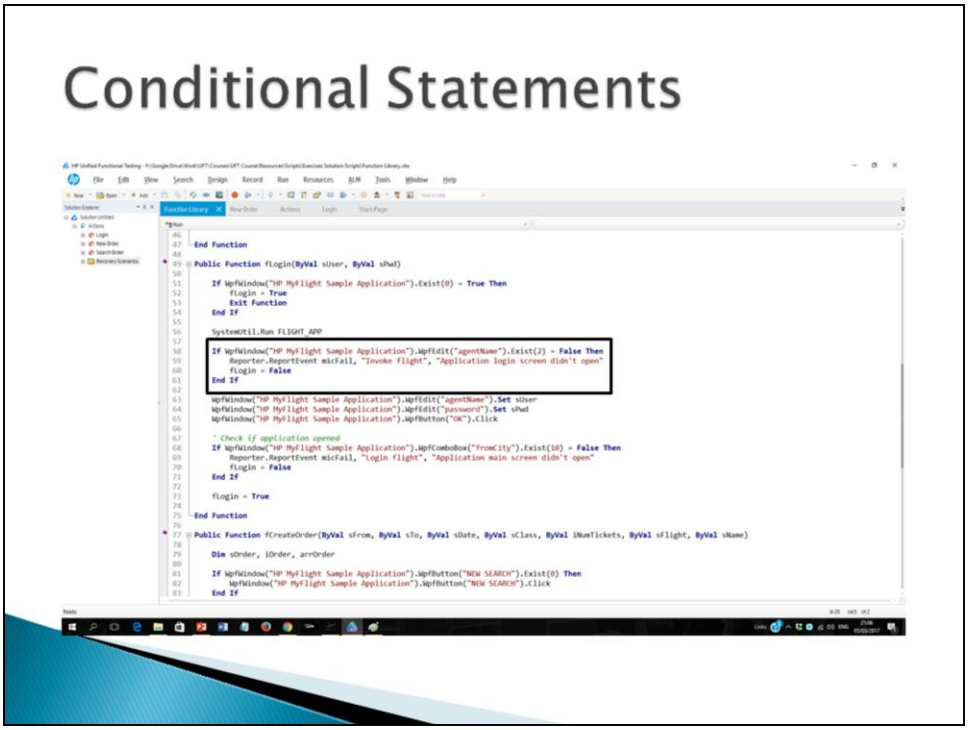
1. From the UFT menu bar, select **FILE → SETTINGS.**
2. In the **TEST SETTINGS** dialog box, click the **RECOVERY** tab**.**
3. Click the **ADD** button. The **ADD RECOVERY SCENARIO** dialog box appears.
4. Click the **BROWSE** button to locate and open the recovery scenario file.
5. In the **ADD RECOVERY SCENARIO** dialog box, click **ADD SCENARIO.**

Viewing Test Result



Handling Exception Programmatically

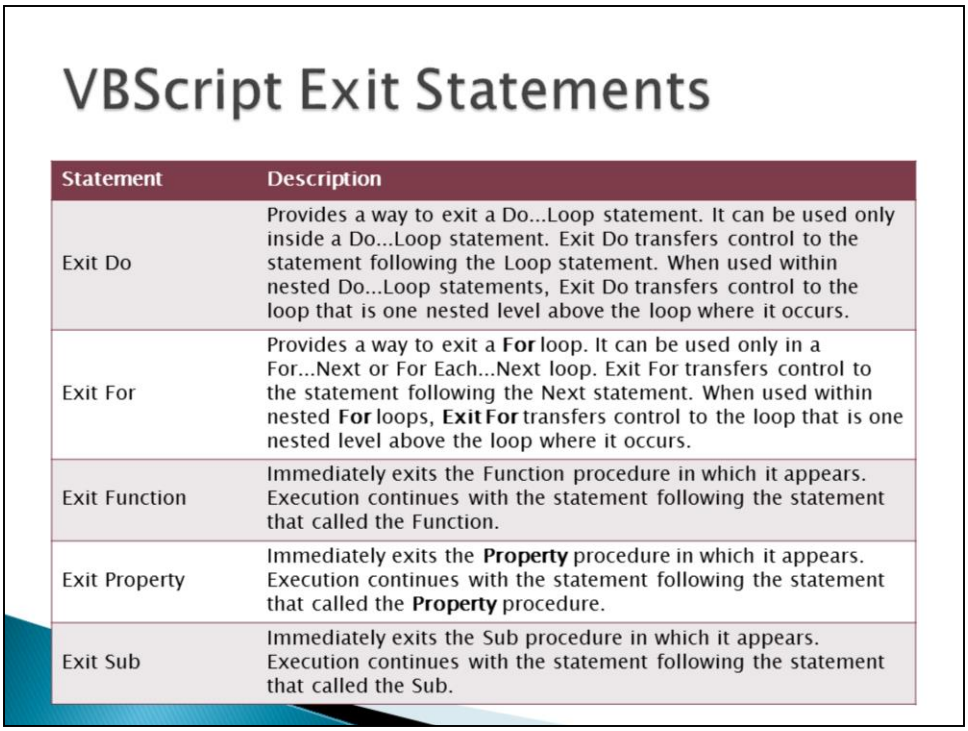
- In addition to handling exceptions by using recovery scenarios, you can handle
- exceptions programmatically for better control over the script. You can handle exceptions programmatically using:
  - Conditional statements.
  - Exit statements.
  - The On Error Resume Next statement.



conditional statements separate the code to be executed for positive or negative test data. The expected result for each UserName and Password combination is stored in the DataTable.

Types Of Exit Statements

- There are two types of Exit statements:
  - VBScript Exit Statement
  - UFT Exit statements

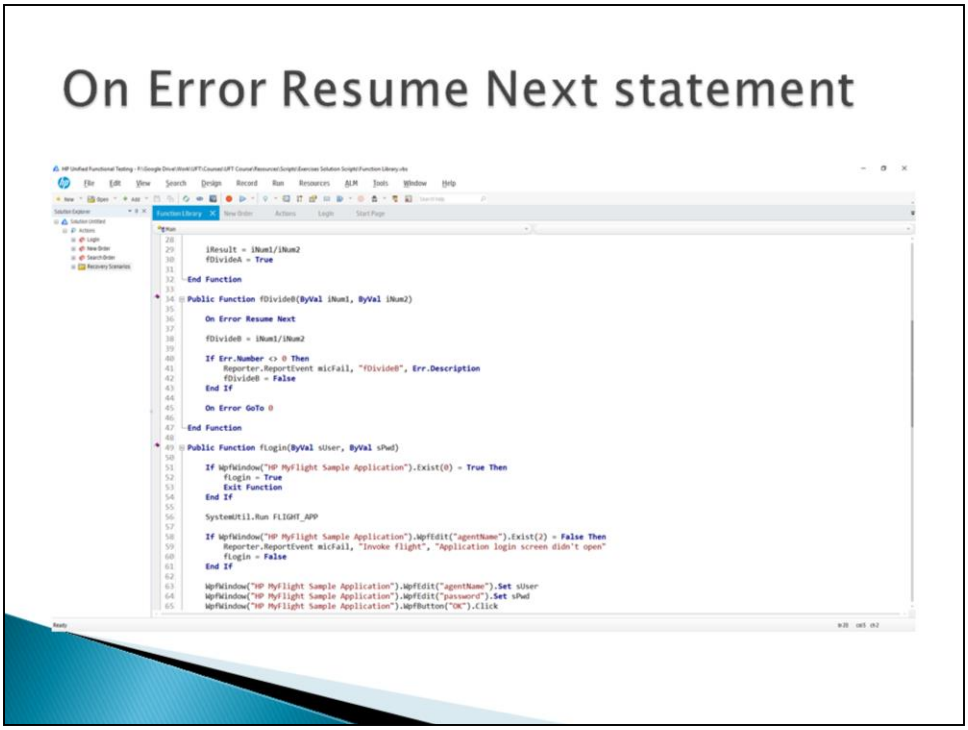


Example:

```
Function RandomLoop
  Dim I, MyNum
  Do   ' Set up infinite loop.
     For I = 1 To 10   ' Loop 1000 times.
        MyNum = Int(Rnd * 10)   ' Generate random numbers.
        Select Case MyNum   ' Evaluate random number.
           Case 1: MsgBox "Case 1"
              Exit For   ' If 1, exit For...Next.
           Case 7: MsgBox "Case 7"
              Exit Do   ' If 7, exit Do...Loop.
           Case 5: MsgBox "Case 5"
              Exit Function   ' If 5, exit Function
           End Select
     Next
  Loop
End Function
```

Types of UFT Exit Statements

UFT Exit Statements	Description
ExitActionIteration	Terminates the current iteration of the action.Proceeds to the next action.
ExitAction	Terminates the current action regardless of the existence of additional iterations.Proceeds to the next action.
ExitGlobalIteration	Terminates the current iteration of the actions in a test. Proceeds to the next iteration.
ExitRun	Terminates the entire run of the test.



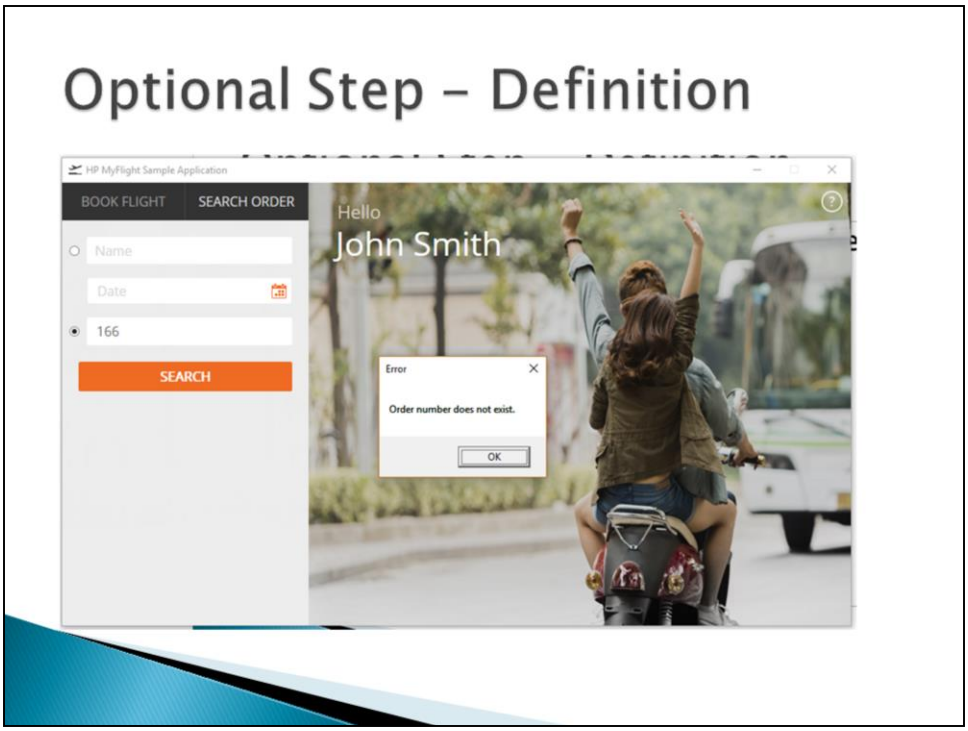
The VBScript On Error Resume Next statement enables a script to continue processing even after an exception occurs. When a script encounters an exception, it simply ignores the current line and proceeds to the next line.

Use the On Error Resume Next statement to enable cleanup code to run entirely, even if exceptions occur.

Note: The On Error Resume Next statement skips all the exceptions in a test including

the ones that you do not anticipate.

Check the value of Err.Number to see if an error occurred.



An optional step is a step that is not necessarily required to successfully complete a run session.

During a run session, if the object of an optional step does not exist, UFT bypasses this step and continues to run the test. When the run session ends, a message is displayed for the step indicating that the step was not performed, but the step does not cause the run to fail.

Setting Optional Steps

- To set an optional step in the **Keyword View**, right-click a step and choose **Optional Step**. The Optional Step icon is added next to the selected step.
- To add an optional step in the **Expert View**, add **OptionalStep** to the beginning of the VBScript statement.

For example:

OptionalStep.*Browser("Browser").Dialog("AutoComplete").WinButton("Yes").Click*

What's Next?

- Review Questions
- Exercise
- Next Lesson
  - The next lesson in the course is:
    Dynamic Data

End of Lesson