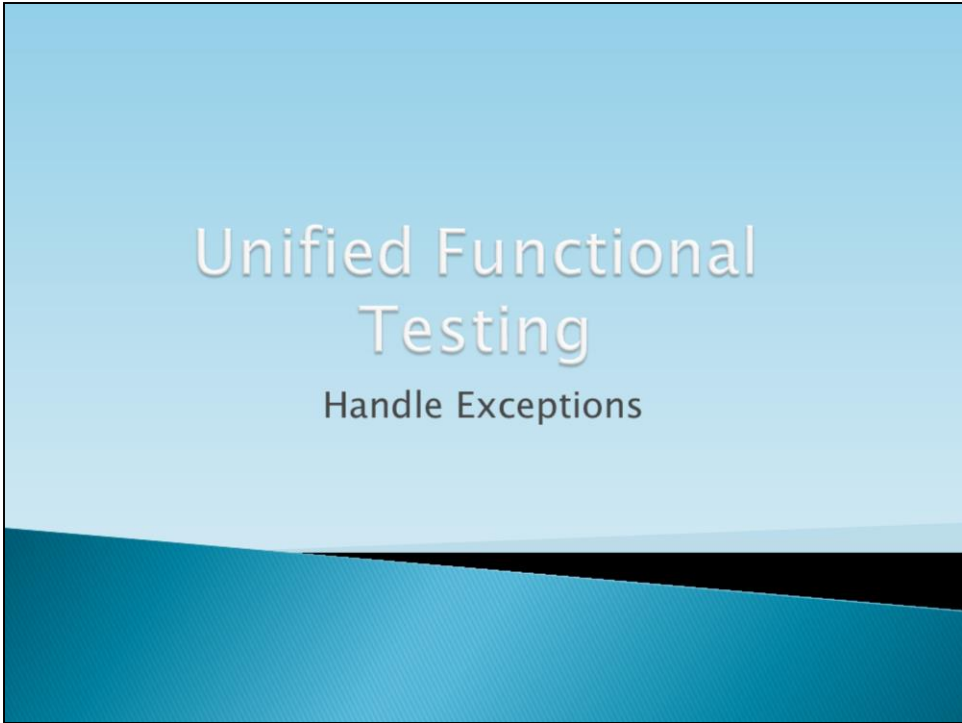


# Unified Functional Testing

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

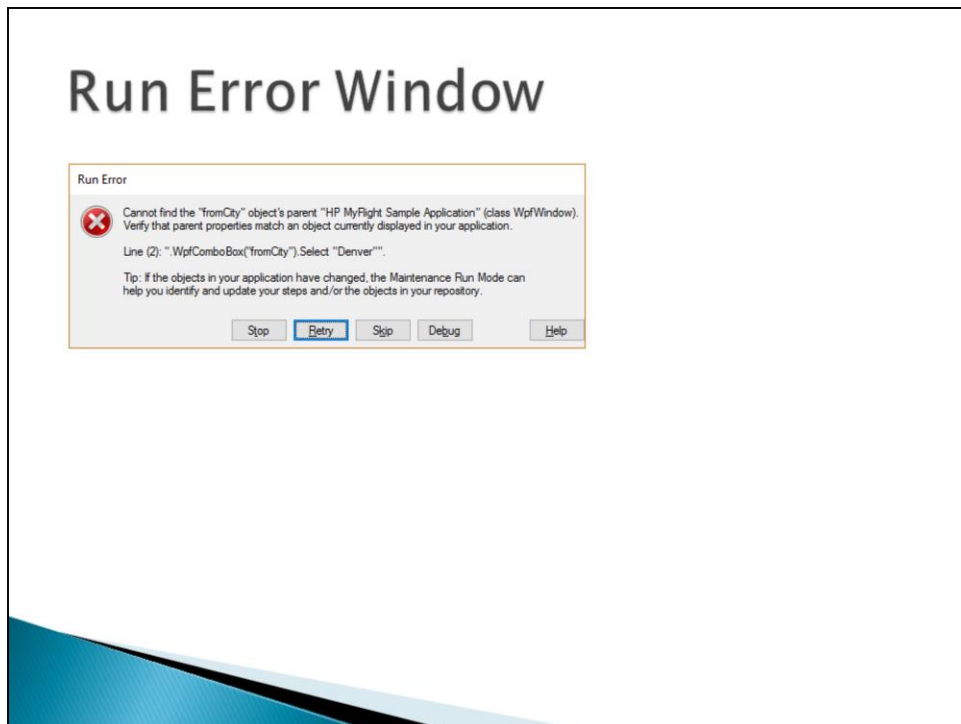
1. What is an exception
2. Recovery manager
3. Exit statements
4. VBScript error object



## 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.





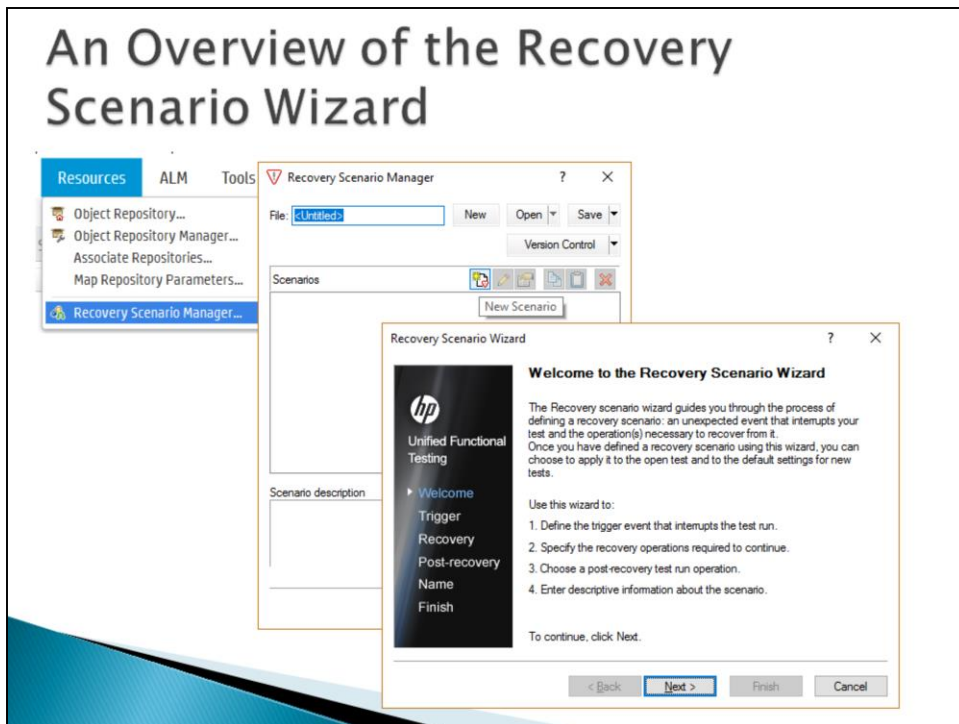
When the UFT encounters an exception within a run session, it displays a RUN ERROR window. The RUN ERROR window provides the following options to handle the exceptions:

- **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.

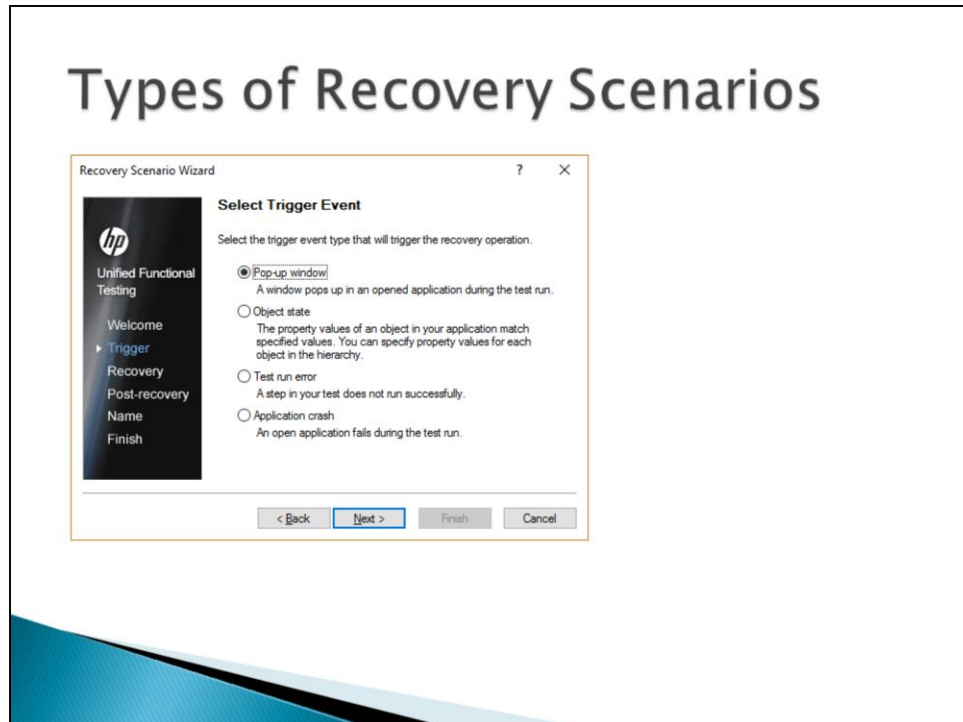
To open RECOVERY SCENARIO WIZARD:

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.



# Specifying the Trigger Conditions

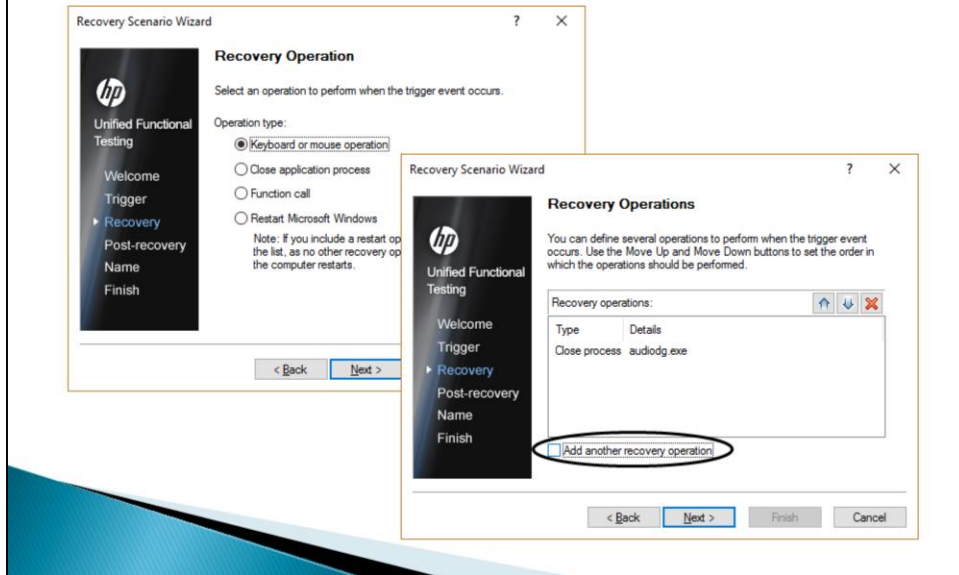
The screenshot shows a window titled 'Recovery Scenario Wizard' with a sidebar on the left containing the following items: 'hp Unified Functional Testing', 'Welcome', 'Trigger' (highlighted with a blue arrow), 'Recovery', 'Post-recovery', 'Name', and 'Finish'. The main area is titled 'Specify Pop-up Window Conditions'. It contains the following text: 'Specify the conditions to identify a pop-up window for this recovery scenario. You can use the pointing hand to capture the title and text values of the pop-up window.' Below this is a 'Condition to identify window:' section with two checked options: 'Window title: Notification' and 'Window text contains: &Yes&NoAre you sure you want to delete this order?'. Each option has a 'Regular expression' checkbox and a 'P' button. At the bottom are '< Back', 'Next >', 'Finish', and 'Cancel' buttons.

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.

# Selecting Recovery Operation

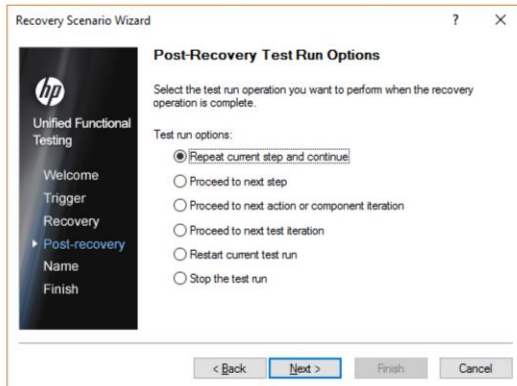


After you specify the trigger event type and the trigger event conditions in **RECOVERY SCENARIO WIZARD**, you specify the recovery operation(s) required to handle the trigger event.

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.

# Selecting The Post-Recovery Operation



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.

## Providing Recovery Scenario Name

The screenshot shows a 'Recovery Scenario Wizard' window with a sidebar on the left containing the HP logo and a list of steps: Unified Functional Testing, Welcome, Trigger, Recovery, Post-recovery, Name (selected), and Finish. The main area is titled 'Name and Description' and contains the instruction 'Provide a name and description for your recovery scenario.' It features three input fields: 'Recovery file:' with the value '<New Recovery Scenario File>', 'Scenario name:' with the value 'PopUp', and a larger 'Description:' text area. At the bottom are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

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

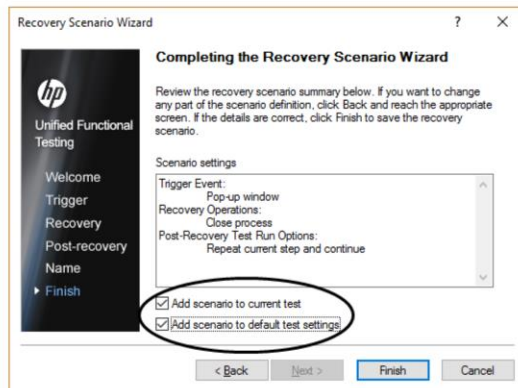
for a recovery scenario, you use the NAME AND DESCRIPTION page to provide a name

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.

# Completing The Recovery Scenario Wizard



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.

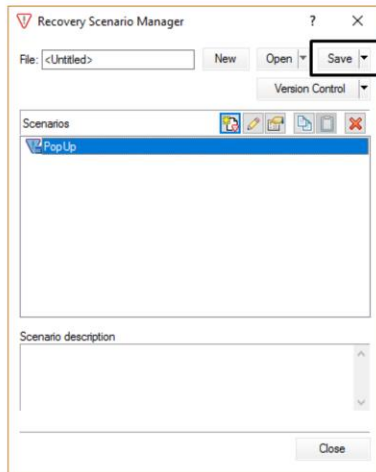
2. To associate the recovery scenario with all tests that you create in UFT, check

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

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

dialog box appears.

# Saving a Recovery Scenario

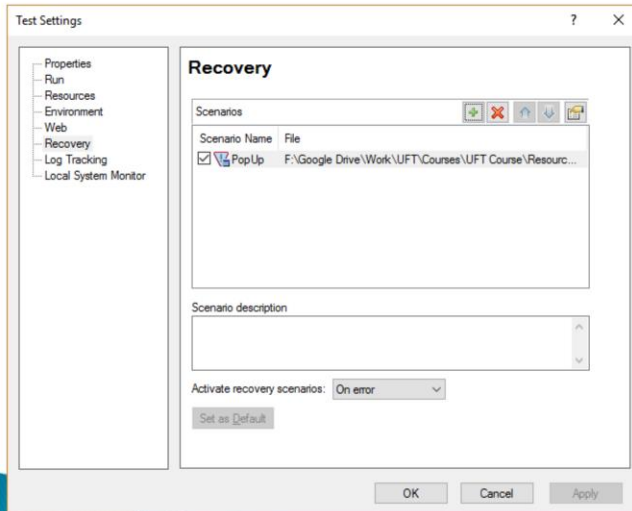


After you create a recovery scenario, save the recovery scenario to a file.

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.

## Viewing the Recovery Scenarios for a Test



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

The screenshot displays the HP UFT Report interface for a test titled "Handle Exceptions - Recovery Scenario - TempResults". The interface includes a top navigation bar with tabs for "UFT Report", "Handle Exceptions - Recovery Scenario - TempResults", "Handle Exceptions - Recovery Scenario", "Action1", and "Start Page". Below the navigation bar, the test title is shown with a green checkmark icon and a status bar indicating 0 errors, 0 warnings, and 0 failures. The execution time is 2017-03-04 12:38:14, and the duration is 00:00:05. The test name is "HP Unified Functional Testing 12.54".

The main content area is divided into two sections: "Errors list" and "Test flow". The "Errors list" section is currently empty. The "Test flow" section shows a sequence of steps: "Test Iteration: Run 1", "Action1", "Nottom", "RecoveryFunction1", "Custom: Recovery", and "Skip". The "Nottom" step is highlighted with a red box. The "Step Details" section on the right provides information about the selected step, including its name, description, and the current state of the test.

**Step Details:**

- Step:** Nottom
- Description:** Scenario: Nottom  
Defined in: F:\Single Drive\Work\UFT\Course\UFT Course\Recovery\Scripts\Exercises Solution\Script\MyRec.ign  
Description: Post recovery operation: Proceed to next step.
- Activated by trigger:** [Error] Test run error  
The error string: Item is list or menu not found
- The current test step details:**  
Object: WinComboBox("NameCity")  
Method: Select  
Arguments: DenverRoads.com  
Result: Item is list or menu not found
- Execution Time:**

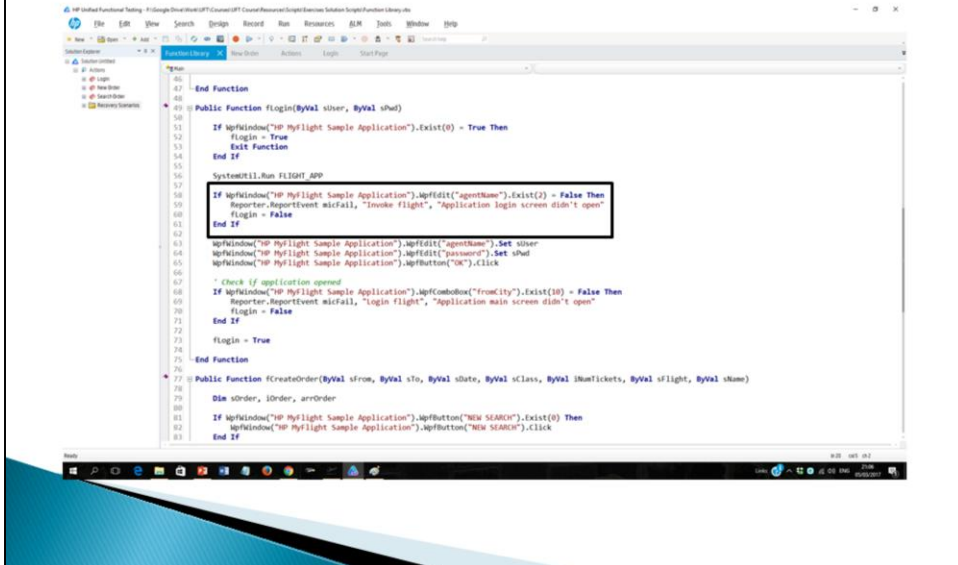


## Handling Exception Programmatically

- ▶ 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



```
40 End Function
41
42 Public Function flogin(ByVal sUser, ByVal sPwd)
43
44     If HpFindWindow("HP MyFlight Sample Application").Exist(0) = True Then
45         flogin = True
46         Exit Function
47     End If
48
49     SystemUtil.Run FLIGHT_APP
50
51     If HpFindWindow("HP MyFlight Sample Application").HpEdit("agentName").Exist(2) = False Then
52         Reporter.ReportEvent micFail, "Invoke flight", "Application login screen didn't open"
53         flogin = False
54     End If
55
56     HpFindWindow("HP MyFlight Sample Application").HpEdit("agentName").Set sUser
57     HpFindWindow("HP MyFlight Sample Application").HpEdit("password").Set sPwd
58     HpFindWindow("HP MyFlight Sample Application").HpButton("OK").Click
59
60     ' Check if application opened
61     If HpFindWindow("HP MyFlight Sample Application").HpComboBox("fromCity").Exist(10) = False Then
62         Reporter.ReportEvent micFail, "login flight", "Application main screen didn't open"
63         flogin = False
64     End If
65
66     flogin = True
67 End Function
68
69 Public Function fCreateOrder(ByVal sFrom, ByVal sTo, ByVal sDate, ByVal sClass, ByVal sNumTickets, ByVal sFlight, ByVal sName)
70
71     Dim sOrder, iOrder, arrOrder
72
73     If HpFindWindow("HP MyFlight Sample Application").HpButton("NEW SEARCH").Exist(0) Then
74         HpFindWindow("HP MyFlight Sample Application").HpButton("NEW SEARCH").Click
75     End If
76 End Function
```

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



# VBScript Exit Statements

Statement	Description
Exit Do	Provides a way to exit a Do...Loop statement. It can be used only inside a Do...Loop statement. Exit Do transfers control to the statement following the Loop statement. When used within nested Do...Loop statements, Exit Do transfers control to the loop that is one nested level above the loop where it occurs.
Exit For	Provides a way to exit a For loop. It can be used only in a For...Next or For Each...Next loop. Exit For transfers control to the statement following the Next statement. When used within nested For loops, Exit For transfers control to the loop that is one nested level above the loop where it occurs.
Exit Function	Immediately exits the Function procedure in which it appears. Execution continues with the statement following the statement that called the Function.
Exit Property	Immediately exits the Property procedure in which it appears. Execution continues with the statement following the statement that called the Property procedure.
Exit Sub	Immediately exits the Sub procedure in which it appears. Execution continues with the statement following the statement that called the Sub.

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
```

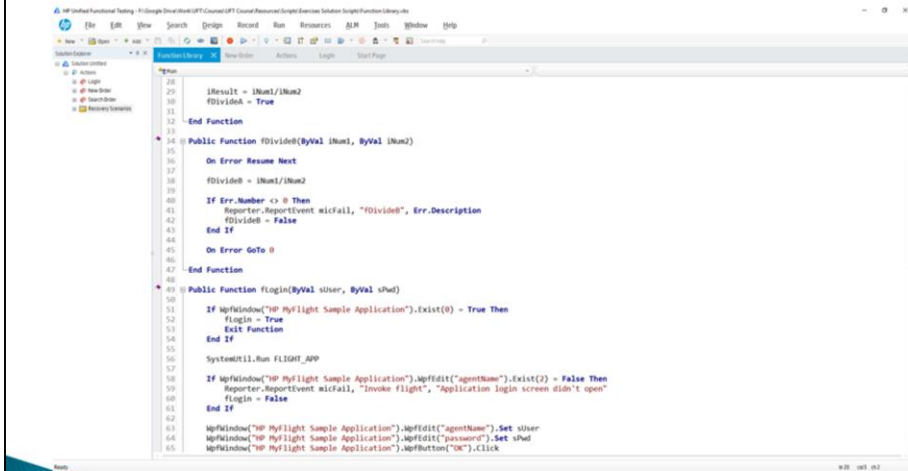
```
Loop
```

```
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.

# On Error Resume Next statement



```
28 iResult = iNum1/iNum2
29 fDivideb = True
30
31 -End Function
32
33
34 @ Public Function fDivideb(ByVal iNum1, ByVal iNum2)
35
36     On Error Resume Next
37     fDivideb = iNum1/iNum2
38
39     If Err.Number <> 0 Then
40         Reporter.ReportEvent micFail, "fDivideb", Err.Description
41         fDivideb = False
42     End If
43
44     On Error GoTo 0
45
46 -End Function
47
48
49 @ Public Function fLogin(ByVal sUser, ByVal sPwd)
50
51     If UpfWindow("HP MyFlight Sample Application").Exist(0) = True Then
52         fLogin = True
53         Exit Function
54     End If
55
56     SystemUtil.Run FLIGHT_APP
57
58     If UpfWindow("HP MyFlight Sample Application").AppfEdit("agentName").Exist(2) = False Then
59         Reporter.ReportEvent micFail, "Invoke flight", "Application login screen didn't open"
60         fLogin = False
61     End If
62
63     UpfWindow("HP MyFlight Sample Application").AppfEdit("agentName").Set sUser
64     UpfWindow("HP MyFlight Sample Application").AppfEdit("password").Set sPwd
65     UpfWindow("HP MyFlight Sample Application").AppfButton("OK").Click
66 End Function
```

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.

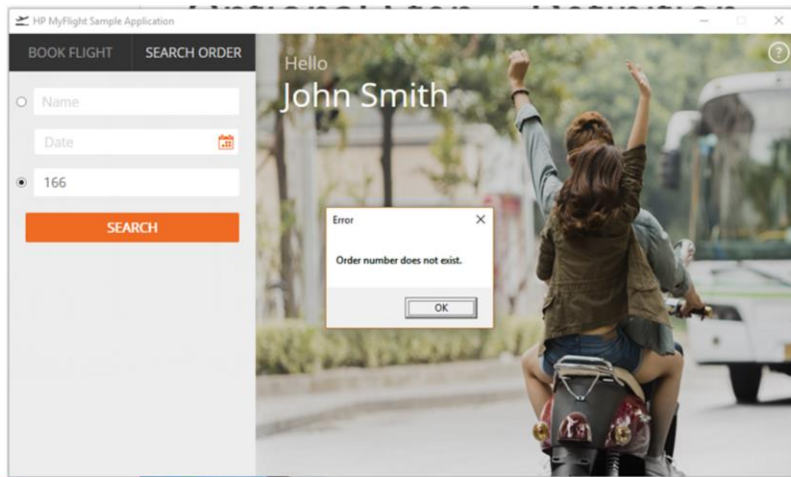
Use the On Error Resume Next statement to enable cleanup code to run entirely, even if exceptions occur.

**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.

## Optional Step – Definition



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

