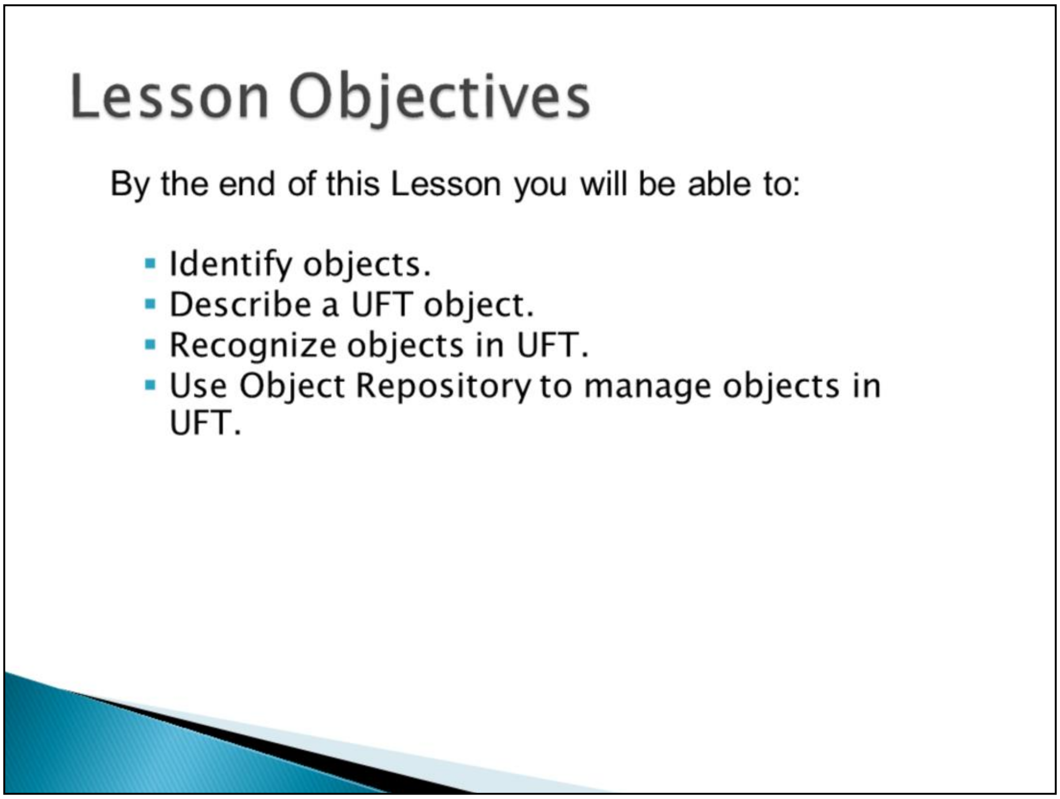
Unified Functional Testing

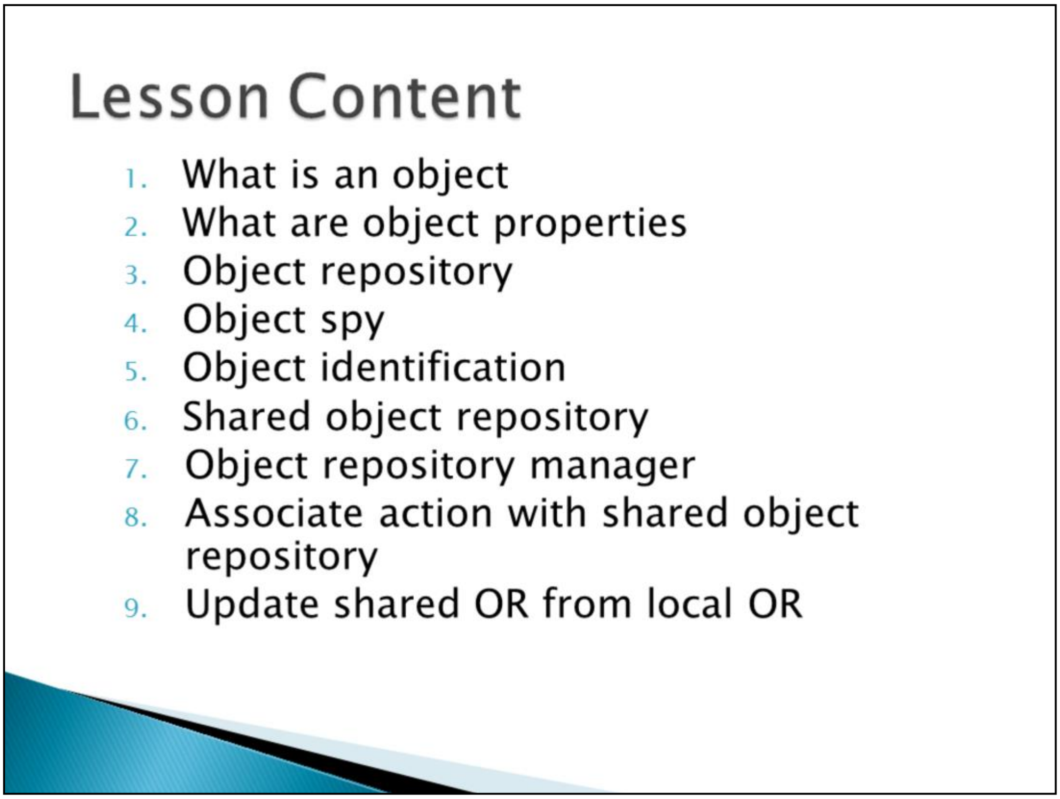
Object Repository



Lesson Objectives

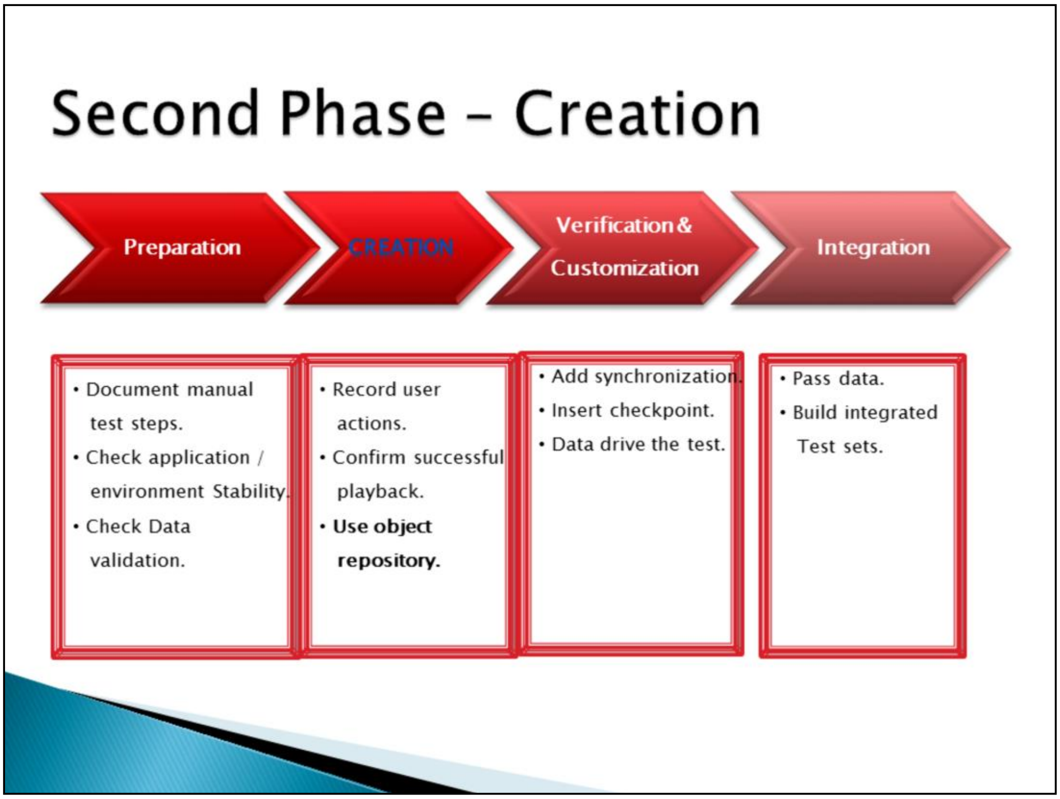
By the end of this Lesson you will be able to:

- Identify objects.
- Describe a UFT object.
- Recognize objects in UFT.
- Use Object Repository to manage objects in UFT.



Lesson Content

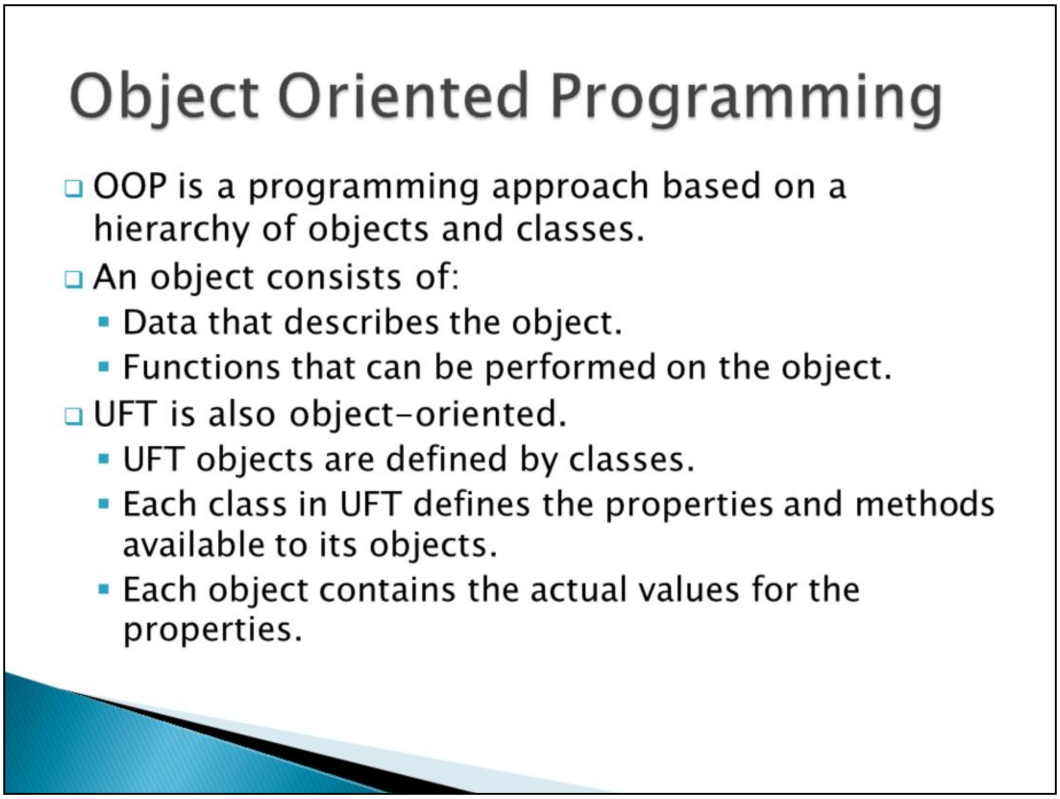
1. What is an object
2. What are object properties
3. Object repository
4. Object spy
5. Object identification
6. Shared object repository
7. Object repository manager
8. Associate action with shared object repository
9. Update shared OR from local OR



Second Phase – Creation

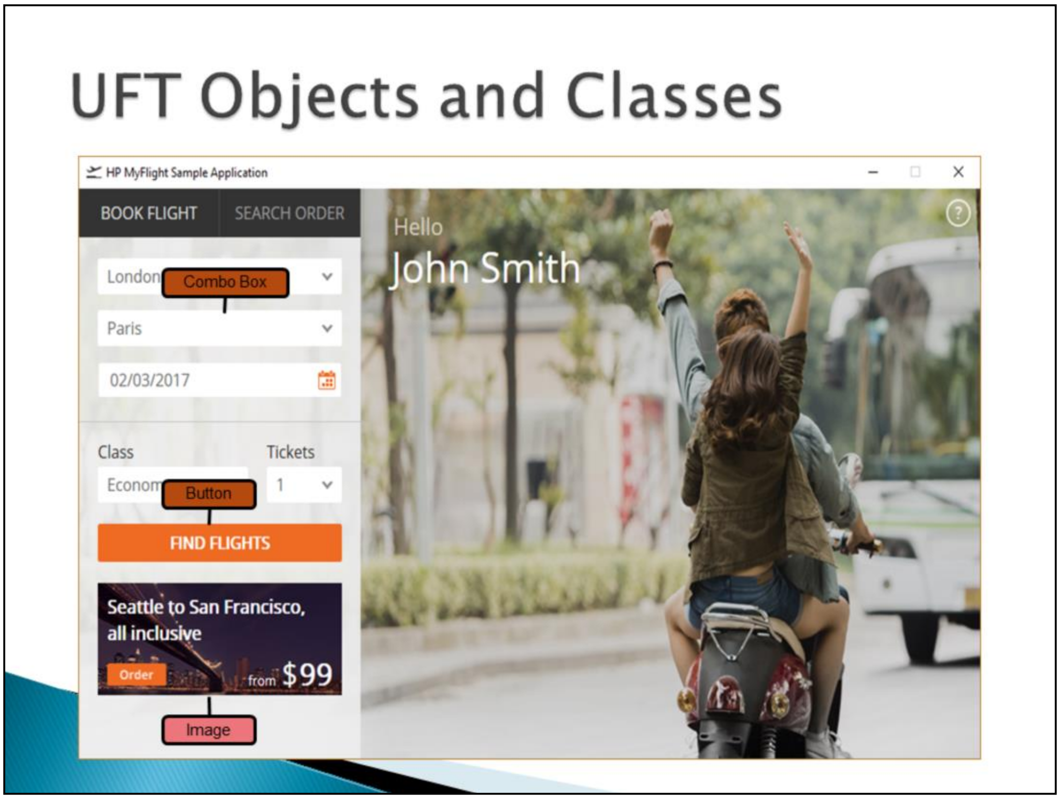
Preparation	CREATION	Verification & Customization	Integration
• Document manual test steps. • Check application / environment Stability. • Check Data validation.	• Record user actions. • Confirm successful playback. • **Use object repository.**	• Add synchronization. • Insert checkpoint. • Data drive the test.	• Pass data. • Build integrated Test sets.

After recording a test, the test object information is stored in the Object Repository.



Object Oriented Programming

- OOP is a programming approach based on a hierarchy of objects and classes.
- An object consists of:
  - Data that describes the object.
  - Functions that can be performed on the object.
- UFT is also object-oriented.
  - UFT objects are defined by classes.
  - Each class in UFT defines the properties and methods available to its objects.
  - Each object contains the actual values for the properties.



UFT Objects and Classes

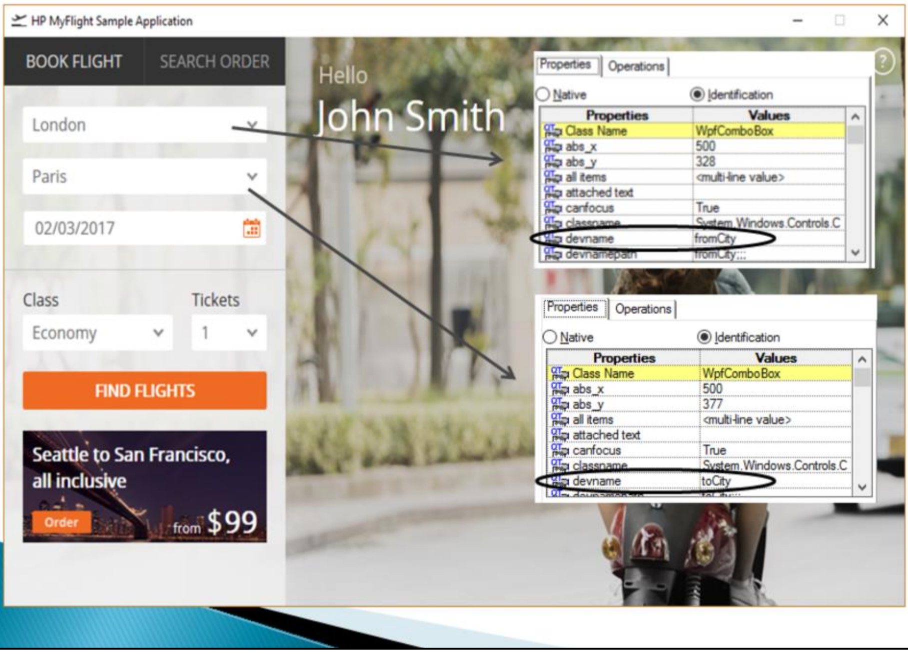
A UFT object is a graphic user element in the AUT.

UFT objects are categorized into classes, such as Button, Edit, ComboBox and Radio Button.

UFT does not define any object information; it uses the same information that application developers defined.

The FROM and TO objects belong to the WPFCOMBOBOX class. However, the properties of the two objects different.

Object Property

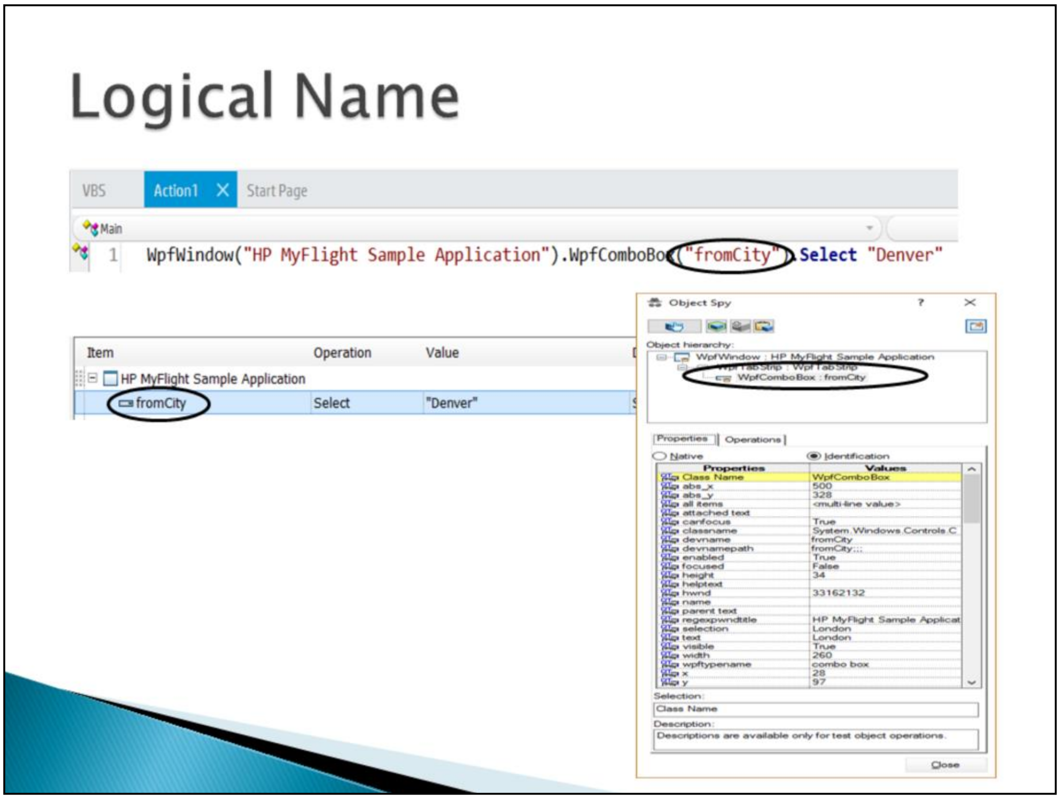


○ Native ● Identification

Properties	Values
Class Name	WpfComboBox
abs_x	500
abs_y	328
all items	<multi-line value>
attached text	
canfocus	True
classname	System.Windows.Controls.C
devname	fromCity
devnamepath	fromCity:::

○ Native ● Identification

Properties	Values
Class Name	WpfComboBox
abs_x	500
abs_y	377
all items	<multi-line value>
attached text	
canfocus	True
classname	System.Windows.Controls.C
devname	toCity



Logical Name

```
WpfWindow("HP MyFlight Sample Application").WpfComboBox("fromCity").Select "Denver"
```

Item	Operation	Value
HP MyFlight Sample Application		
fromCity	Select	"Denver"

WpfWindow : HP MyFlight Sample Application
WpfTabStrip : WpfTabStrip
WpfComboBox : fromCity

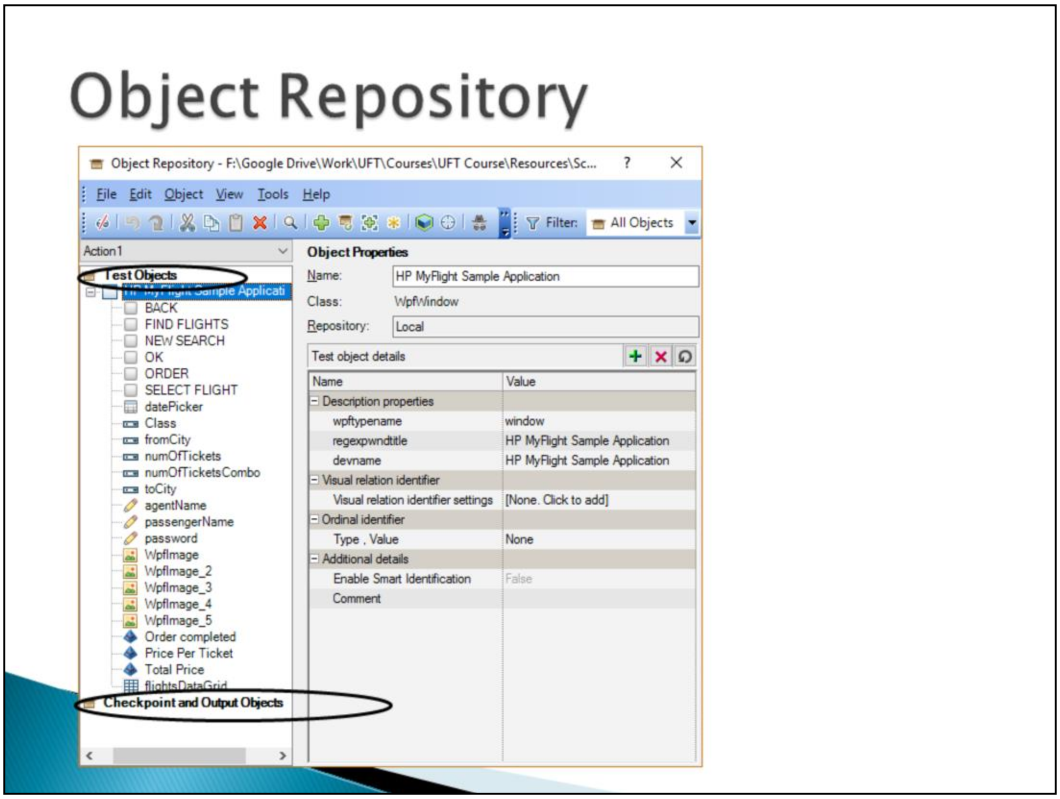
○ Native ● Identification

Properties	Values
Class Name	WpfComboBox
abs_x	500
abs_y	328
all items	<multi-line value>
attached text	
canfocus	True
classname	System.Windows.Controls.C
devname	fromCity
devnamepath	fromCity;;;
enabled	True
focused	False
height	34
helptext	
hwnd	33162132
name	
parent text	
regexpwndtitle	HP MyFlight Sample Applicat
selection	London
text	London
visible	True
width	260
wpftypename	combo box
x	28
y	97

Selection: Class Name

Description: Descriptions are available only for test object operations.

UFT assigns a name to the object. This name is called the logical name of the object.



Object Repository

Object Properties

Name: HP MyFlight Sample Application

Class: WpfWindow

Repository: Local

Test object details

Name	Value
⊟ Description properties	
wpftypename	window
regexpwndtitle	HP MyFlight Sample Application
devname	HP MyFlight Sample Application
⊟ Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
⊟ Ordinal identifier	
Type , Value	None
⊟ Additional details	
Enable Smart Identification	False
Comment	

The Object Repository window displays a tree of all test objects and all checkpoint and output objects in the current component or in the selected action.

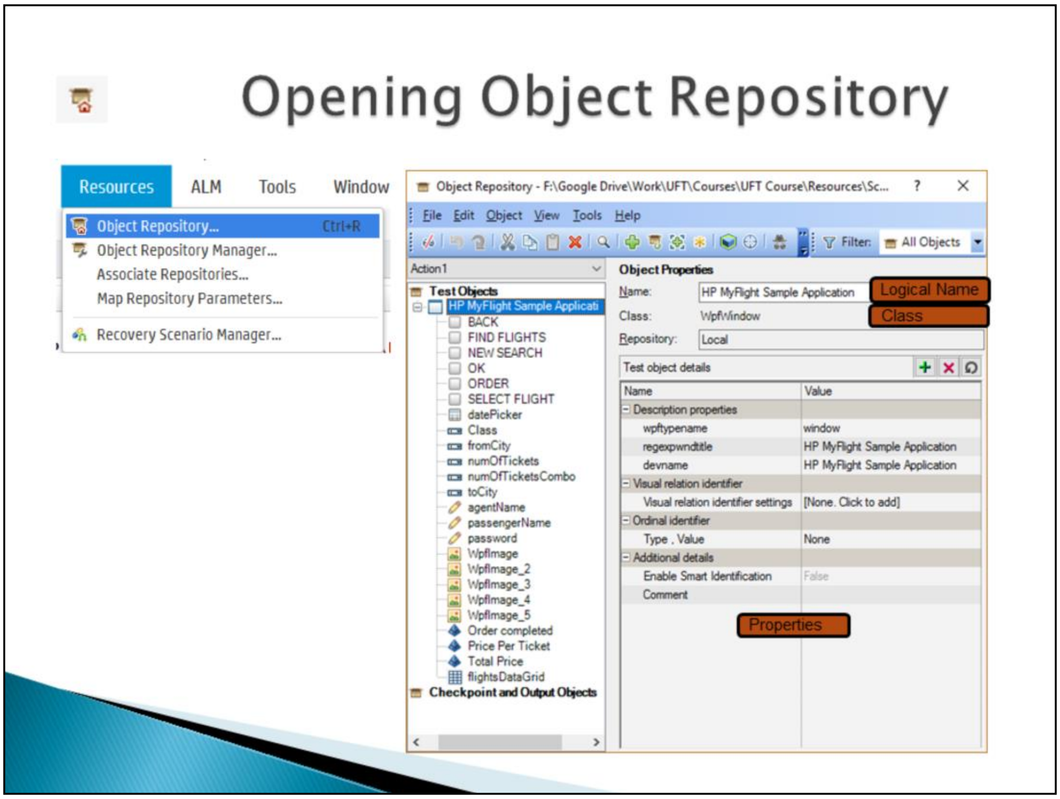
UFT identifies each object that you record and stores it as a test object, determining its object class.

For each object class, UFT identifies a default set of properties.

Usually, UFT needs only a few properties to uniquely identify an object within the scope of the window. In the Flight Reservation application, the CLASS NAME and ATTACHED TEXT properties are enough for UFT to uniquely identify an object.

Test objects can be stored in two types of object repositories: shared and local. By default, the test objects are stored in the local object repository. These test objects are associated with a specific action and no other action can access these objects.

You can use the Object Repository window to view the object description of any object in the repository (in local and shared object repositories), to modify local objects and their properties, and to add test objects to your local object repository. You can also drag and drop test objects from the Object Repository window to your test or component. When you drag and drop a test object to your test or component, UFT inserts a step with the default operation for that test object in your test or component. Checkpoint and output objects cannot be dragged and dropped from the Object Repository window.



Opening Object Repository

Object Properties

Name: HP MyFlight Sample Application

Class: WpfWindow

Repository: Local

Test object details

Name	Value
Description properties	
wpftypename	window
regexpwndtitle	HP MyFlight Sample Application
devname	HP MyFlight Sample Application
Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	False
Comment	

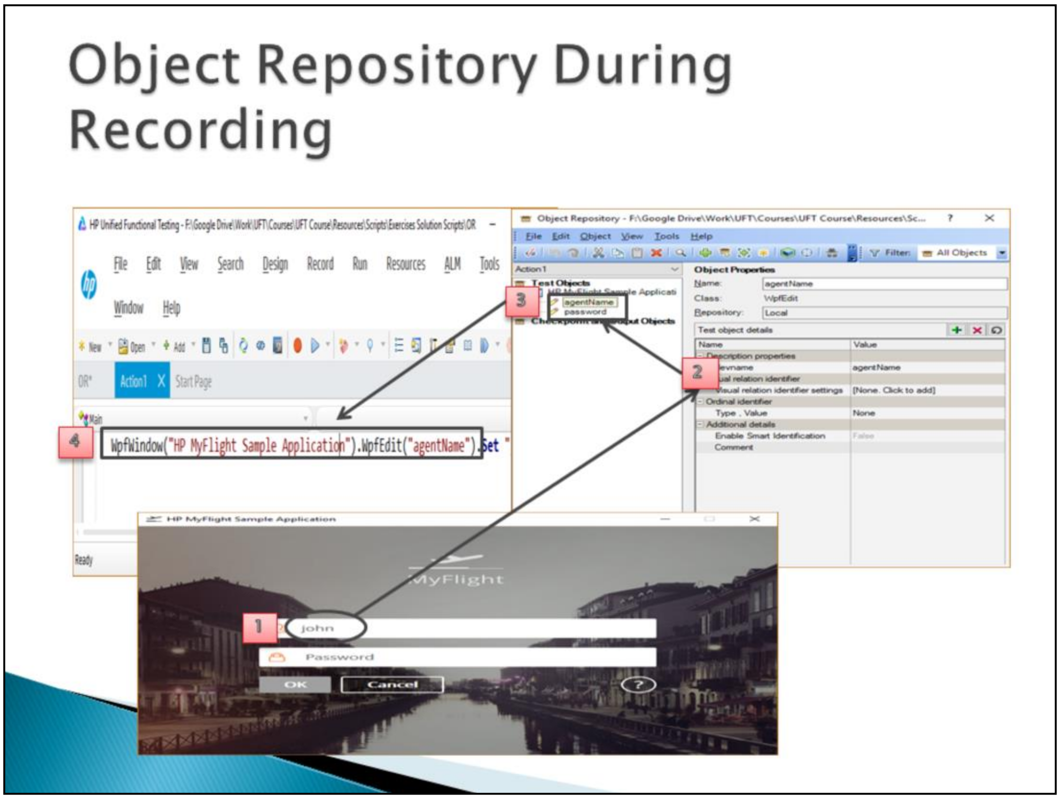
To open the OBJECT REPOSITORY dialog box, in KEYWORD VIEW, select RESOURCES -> OBJECT REPOSITORY.

The OBJECT REPOSITORY displays the following:

•The logical name of an object.

•The class of an object.

•The properties of an object, with the current values



Object Repository During Recording

HP Unified Functional Testing - F:\Google Drive\Work\UFT\Courses\UFT Course\Resources\Scripts\Exercises Solution Scripts\OR

```
WpfWindow("HP MyFlight Sample Application").WpfEdit("agentName").Set "
```

Object Repository - F:\Google Drive\Work\UFT\Courses\UFT Course\Resources\Sc...

Object Properties

Name: agentName

Class: WpfEdit

Repository: Local

Test object details

Name	Value
Description properties	
devname	agentName
Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	False
Comment	

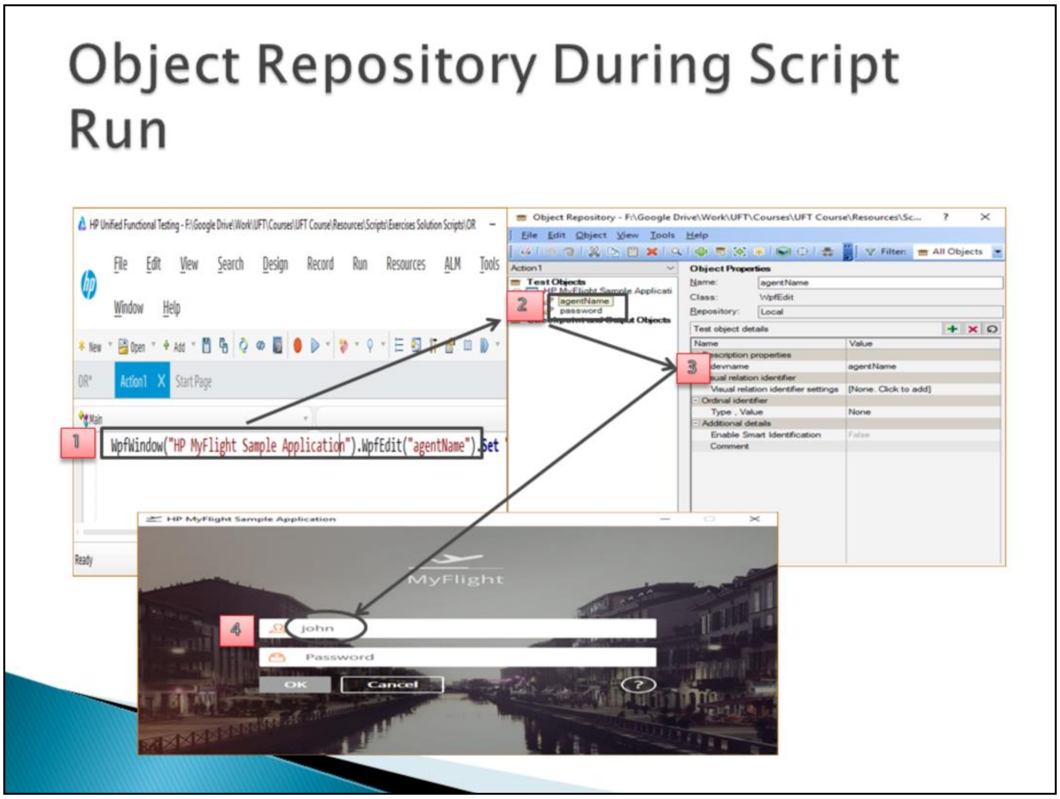
As a test is recorded, every window, button, and other object encountered in the

AUT but not already defined in the Object Repository is added to the Local Object Repository. During recording:

1. UFT gets the properties of the object you are performing the action on and looks for an object with those same physical attributes in the Object Repository.

2. If UFT finds an object with those properties, it uses the logical name for that object in the script.

3. If UFT *doesn’t find an object with those properties in the* Object Repository*, it* creates a new object in the Object Repository, assigns it a logical name, and then uses the logical name in the script.



Object Repository During Script Run

```
WpfWindow("HP MyFlight Sample Application").WpfEdit("agentName").Set
```

Object Properties	
Name:	agentName
Class:	WpfEdit
Repository:	Local

Test object details

Name	Value
Description properties	
devname	agentName
Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	False
Comment	

HP Unified Functional Testing - F:\Google Drive\Work\UFT\Courses\UFT Course\Resources\Scripts\Exercises Solution Scripts\OR
File
Edit
View
Search
Design
Record
Run
Resources
ALM
Tools
Window
Help
New
Open
Add
OR*
Action1
Start Page
Main
Ready
Object Repository - F:\Google Drive\Work\UFT\Courses\UFT Course\Resources\Sc...
File
Edit
Object
View
Tools
Help
Filter:
All Objects
Action1
Test Objects
HP MyFlight Sample Applicati
agentName
password
Checkpoint and Output Objects
HP MyFlight Sample Application
MyFlight
john
Password
OK
Cancel
1
2
3
4

During a script run, every object referred to by the script must exist in the Object Repository so that UFT can find the physical properties for each object and successfully execute

actions against the correct objects in the AUT. Object Repository containing objects

referenced by the script must be associate before running the script.

The Object Repository translates logical names into physical descriptions. There is no link

between the object names appearing in the script and the objects in the AUT other than

the one created by UFT in the Object Repository. All communication between UFT and the AUT passes through the Object Repository.

The process is:

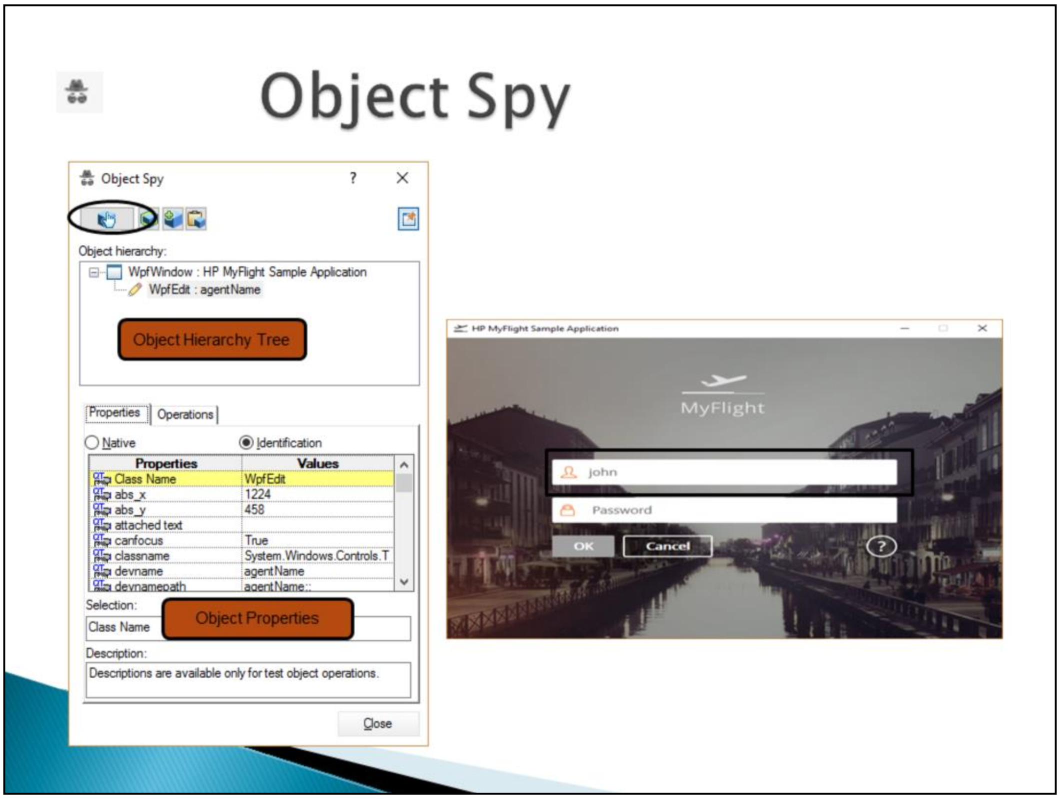
1. Is there a logical name in the Object Repository in the same window that matches the logical name in the script? If there is, get the physical description attached to that logical

name.

2. Is there an object in the AUT, currently open or active, matching that physical

description?

3. If there is, perform the action indicated by the script statement on that object.



Object Spy

Properties	Values
Class Name	WpfEdit
abs_x	1224
abs_y	458
attached text	
canfocus	True
classname	System.Windows.Controls.T
devname	agentName
devnamepath	agentName::

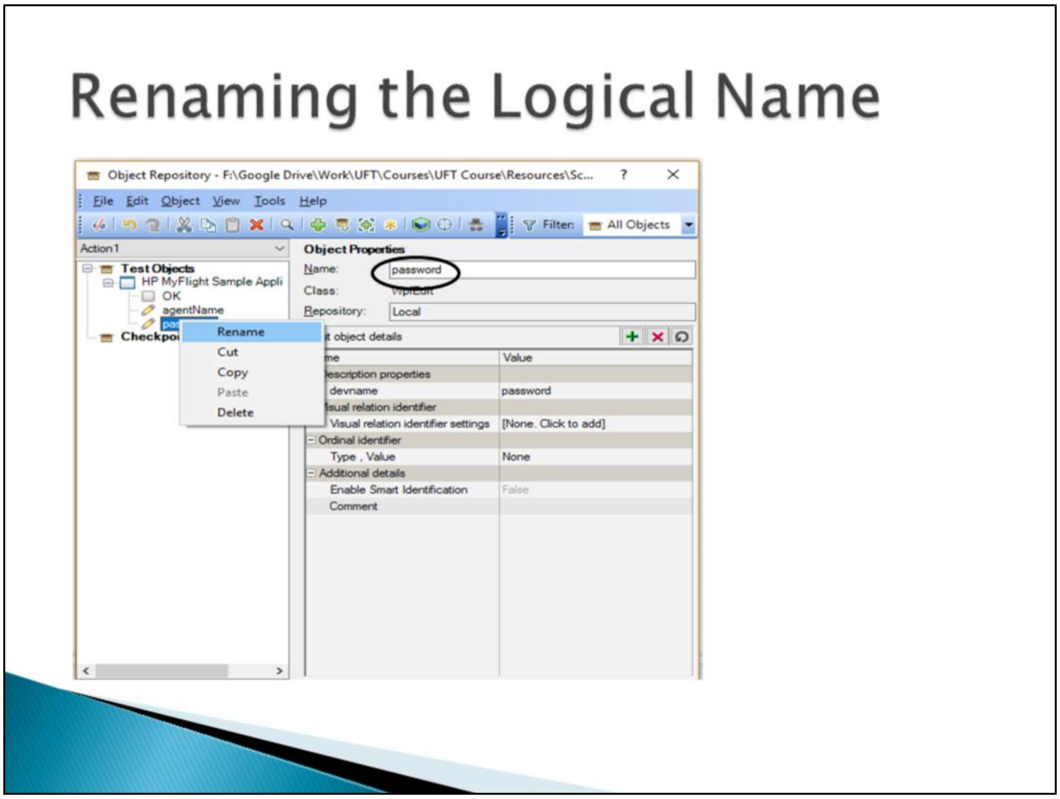
Selection: Class Name

Description: Descriptions are available only for test object operations.

The Object Spy feature enables you to view the properties and their current values as well as methods associated with an object. By pressing on Methods tab , you can view the syntax for a selected method.

In order to open Object Spy

1. select TOOLS -> OBJECT SPY.
2. In the OBJECT SPY dialog box, click the pointing hand button.
3. With the pointing hand, click on the desired object in the AUT.
4. In the OBJECT SPY dialog box, click the **PROPERTIES** tab to view the properties of the selected object.



Renaming the Logical Name

Object Repository - F:\Google Drive\Work\UFT\Courses\UFT Course\Resources\Sc...

Object Properties

Name: password

Class: WpfEdit

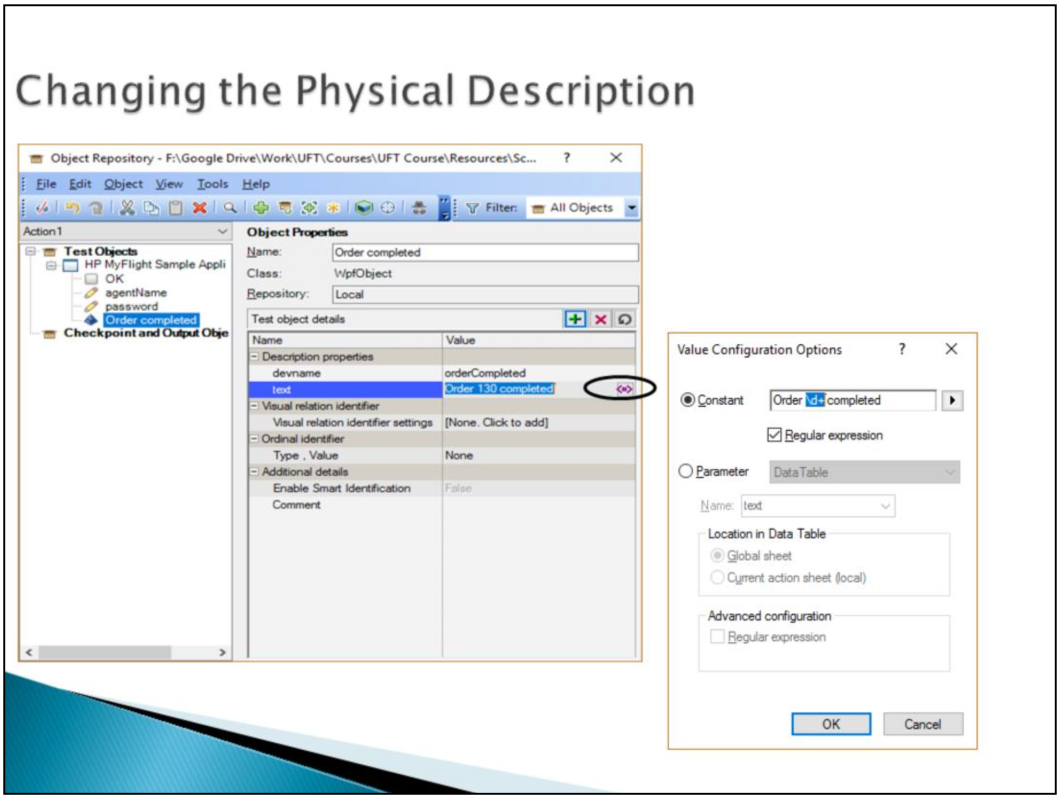
Repository: Local

t object details

me	Value
escription properties	
devname	password
isual relation identifier	
Visual relation identifier settings	[None. Click to add]
⊟ Ordinal identifier	
Type , Value	None
⊟ Additional details	
Enable Smart Identification	False
Comment	

To edit the logical name in OBJECT REPOSITORY dialog box:

1. Right-click the object in the OBJECT REPOSITORY tree.
2. Select **RENAME** from the menu.
3. Type a descriptive name for the object.
4. Select **FILE -> CLOSE.**



Changing the Physical Description

Object Repository - F:\Google Drive\Work\UFT\Courses\UFT Course\Resources\Sc...

Object Properties

Name: Order completed

Class: WpfObject

Repository: Local

Test object details

Name	Value
Description properties	
devname	orderCompleted
text	Order 130 completed
Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	False
Comment	

Value Configuration Options

● Constant: Order \d+ completed

☑ Regular expression

○ Parameter: DataTable

Name: text

Location in Data Table

○ Global sheet

○ Current action sheet (local)

Advanced configuration

□ Regular expression

You should change property value in 2 cases:

1. Property value has changed in the new application version:

Use the “Object Spy” to get the correct value of the property, copy it and replace the old value.

2. Property value is dynamic

Some of the object property’s value contain dynamic parameter, for example the object ‘Fax Order No. 11’ contain number that will be different at the next execution.

You can use regular expression for an object property’s value in Object Repository:

1. Open the **OBJECT REPOSITORY**.

2. From the **OBJECT REPOSITORY** tree, select the object that changes dynamically.

3. In the **TEST OBJECT DETAILS** section, click the **VALUE** field for the property that you

want to modify. The **CONFIGURE THE VALUE** button appears.

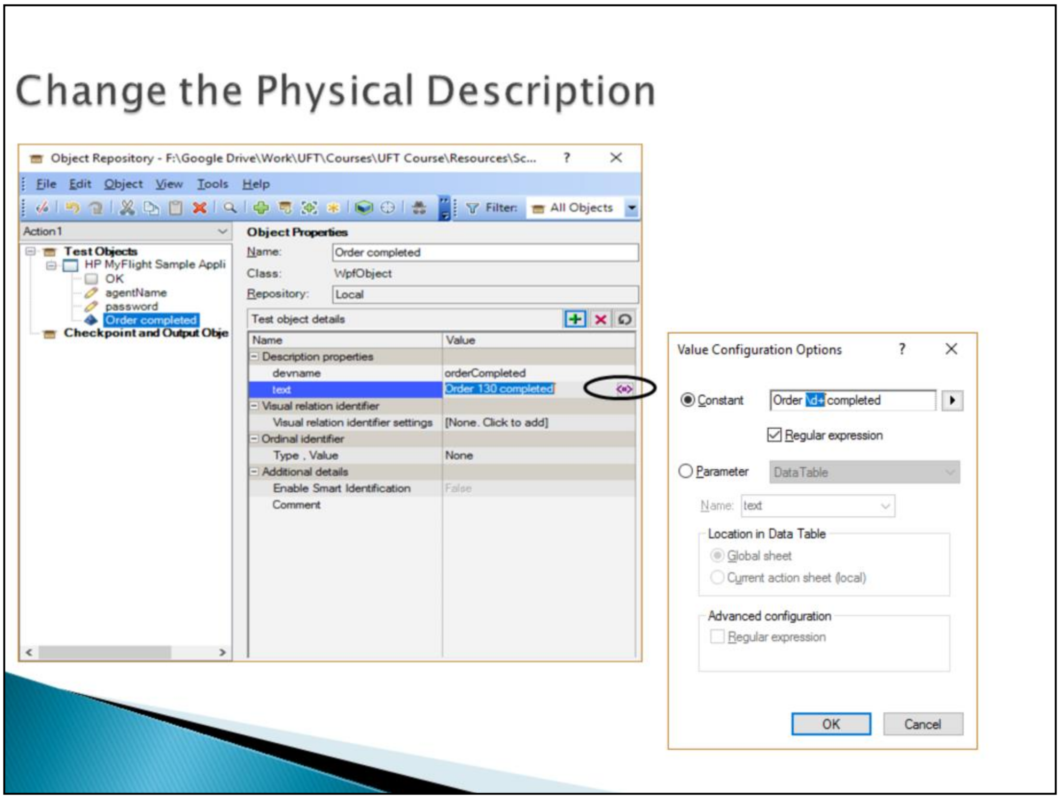
4. Click the **CONFIGURE THE VALUE** button. The **VALUE CONFIGURATION OPTIONS**

dialog box appears.

5. Check the Regular expression checkbox.

6. Write the pattern instead of the constant property value.

Change the Physical Description



Object Repository - F:\Google Drive\Work\UFT\Courses\UFT Course\Resources\Sc...
File Edit Object View Tools Help
Filter: All Objects
Action1
Test Objects
HP MyFlight Sample Appli
OK
agentName
password
Order completed
Checkpoint and Output Obje

Object Properties

Name: Order completed

Class: WpfObject

Repository: Local

Test object details

Name	Value
Description properties	
devname	orderCompleted
text	Order 130 completed
Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	False
Comment	

Value Configuration Options

● Constant: Order \d+ completed

☑ Regular expression

○ Parameter: DataTable

Name: text

Location in Data Table

○ Global sheet

○ Current action sheet (local)

Advanced configuration

□ Regular expression

You should change property value in 2 cases:

1. Property value has changed in the new application version:

Use the “Object Spy” to get the correct value of the property, copy it and replace the old value.

2. Property value is dynamic

Some of the object property’s value contain dynamic parameter, for example the object ‘Fax Order No. 11’ contain number that will be different at the next execution.

You can use regular expression for an object property’s value in Object Repository:

1. Open the **OBJECT REPOSITORY**.

2. From the **OBJECT REPOSITORY** tree, select the object that changes dynamically.

3. In the **TEST OBJECT DETAILS** section, click the **VALUE** field for the property that you

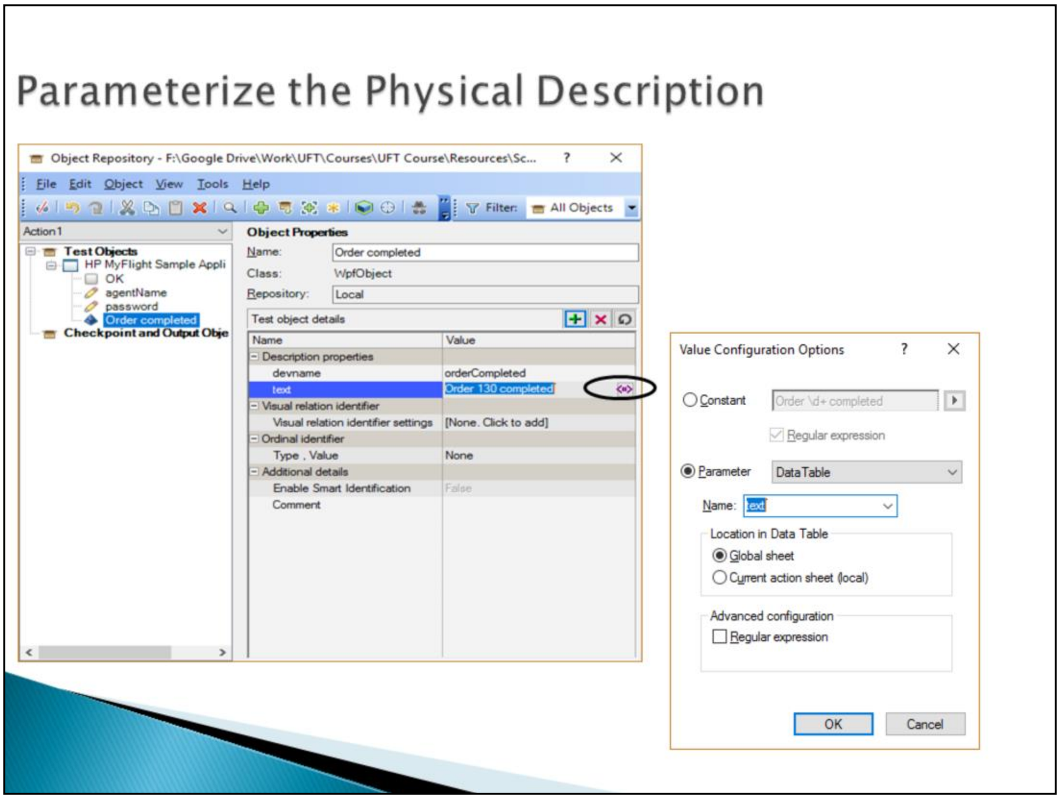
want to modify. The **CONFIGURE THE VALUE** button appears.

4. Click the **CONFIGURE THE VALUE** button. The **VALUE CONFIGURATION OPTIONS**

dialog box appears.

5. Check the Regular expression checkbox.

6. Write the pattern instead of the constant property value.



Parameterize the Physical Description

Object Repository - F:\Google Drive\Work\UFT\Courses\UFT Course\Resources\Sc...

Object Properties

Name: Order completed

Class: WpfObject

Repository: Local

Test object details

Name	Value
Description properties	
devname	orderCompleted
text	Order 130 completed
Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	False
Comment	

Value Configuration Options

○ Constant: Order \d+ completed

☑ Regular expression

● Parameter: DataTable

Name: text

Location in Data Table

● Global sheet

○ Current action sheet (local)

Advanced configuration

☐ Regular expression

You can also parameterize the value to be retrieved from a data table:

In order to parameterize an object property’s value in Object Repository:

1. Open the **OBJECT REPOSITORY**.

2. From the **OBJECT REPOSITORY** tree, select the object that changes dynamically.

3. In the **TEST OBJECT DETAILS** section, click the **VALUE** field for the property that you

want to modify. The **CONFIGURE THE VALUE** button appears.

4. Click the **CONFIGURE THE VALUE** button. The **VALUE CONFIGURATION OPTIONS**

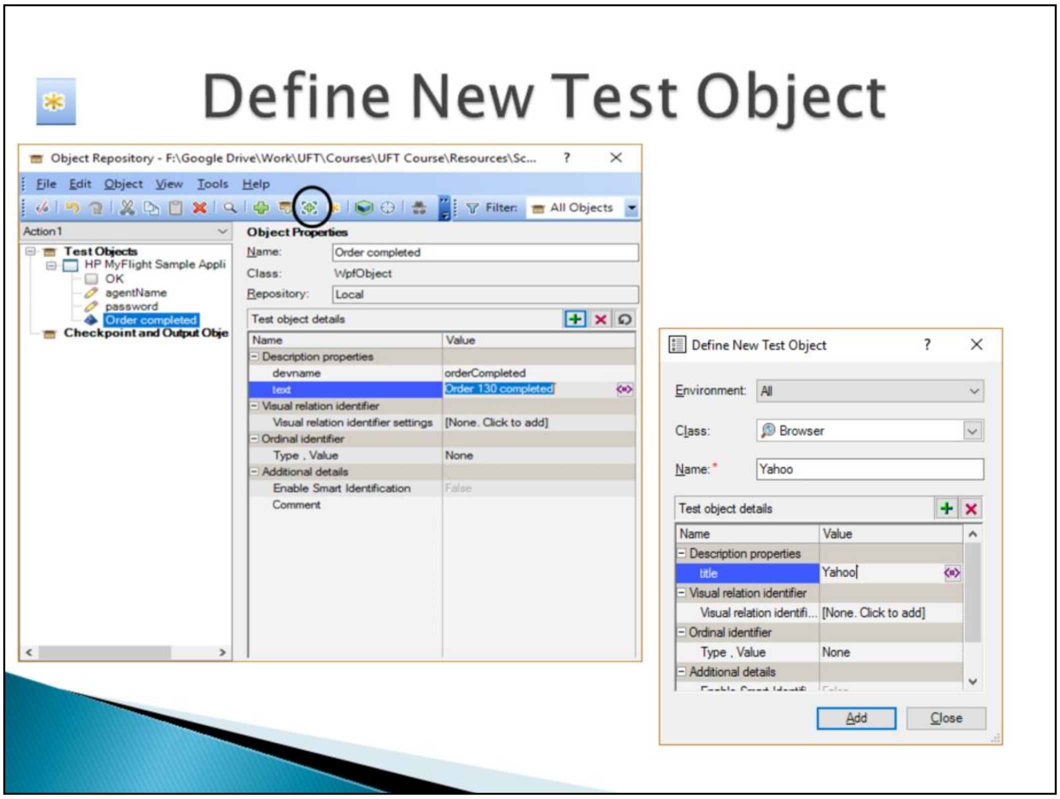
dialog box appears.

5. Select the **PARAMETER** option. From the **PARAMETER** list, select the column name

that stores the dynamically changing object value, and click **OK** to close the **VALUE**

CONFIGURATION OPTIONS dialog box.

6. Close the **OBJECT REPOSITORY**.



Define New Test Object

Object Repository - F:\Google Drive\Work\UFT\Courses\UFT Course\Resources\Sc...

Object Properties

Name: Order completed

Class: WpfObject

Repository: Local

Test object details

Name	Value
Description properties	
devname	orderCompleted
text	Order 130 completed
Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	False
Comment	

Define New Test Object

Environment: All

Class: Browser

Name: * Yahoo

Test object details

Name	Value
Description properties	
title	Yahoo
Visual relation identifier	
Visual relation identifi...	[None. Click to add]
Ordinal identifier	
Type , Value	None
Additional details	

You can define test objects in your object repository that do not yet exist in your application. This enables you to prepare an object repository and build tests or components for your application before the application is ready for testing.

To define a new test object:

1. Open the OBJECT REPOSITORY.

2. Expand the OBJECT REPOSITORY tree and select the parent object under which you want to define the new object.

3. On the UFT toolbar, click the **DEFINE NEW TEST OBJECT** button. The DEFINE NEW TEST OBJECT dialog box appears.

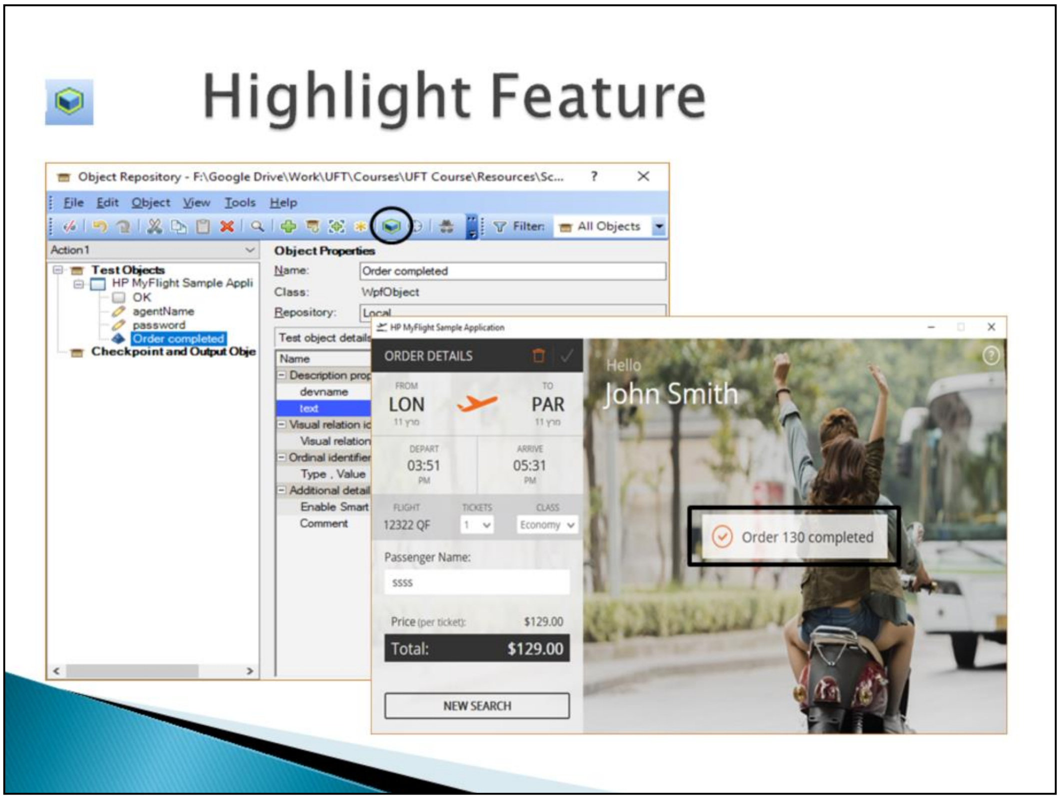
4. From the CLASS list, select the class of the test object.

5. In the NAME field, type a name for the new test object.

6. In the TEST OBJECT DETAILS section, you can specify test object properties and

values for the new object.

7. Click **ADD** to add the new test object to the local object repository.



Highlight Feature

Name: Order completed
Class: WpfObject
Repository: Local

FROM LON 11 מרץ
TO PAR 11 מרץ
DEPART 03:51 PM
ARRIVE 05:31 PM
FLIGHT 12322 QF
TICKETS 1
CLASS Economy
Passenger Name: ssss
Price (per ticket): $129.00
Total: $129.00

Order 130 completed

You can select an object in the OBJECT REPOSITORY dialog box and highlight it in the application you are testing. This is helpful if an object definition has changed and you want to confirm that UFT still recognizes the object.

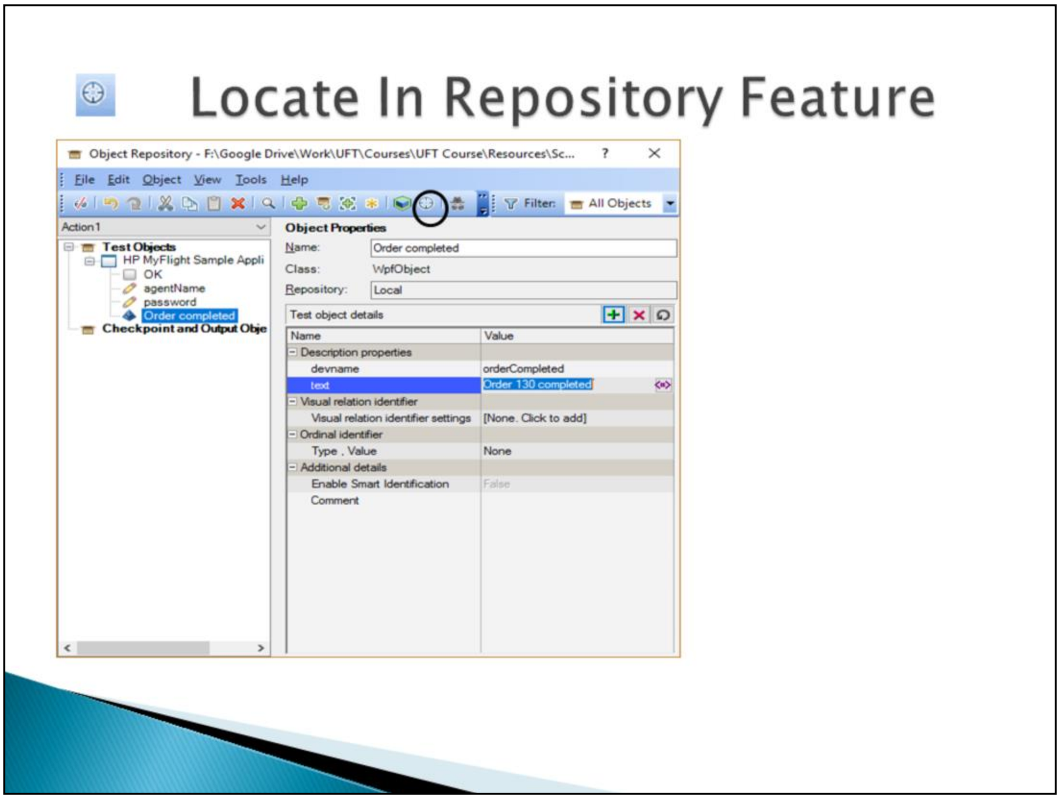
To highlight the object in the application:

Before using this feature, ensure that the application and the specific page are open.

1. Open the OBJECT REPOSITORY dialog box.

2. Select the object in the OBJECT REPOSITORY tree and on the toolbar, click **HIGHLIGHT IN APPLICATION.**

UFT indicates the location of the selected object in your application by temporarily showing a frame around the object and causing it to flash briefly.



Locate In Repository Feature

Object Repository - F:\Google Drive\Work\UFT\Courses\UFT Course\Resources\Sc...

Object Properties

Name: Order completed

Class: WpfObject

Repository: Local

Test object details

Name	Value
Description properties	
devname	orderCompleted
text	Order 130 completed
Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	False
Comment	

You can select an object in the application you are testing and highlight the test object in the object repository.

To locate an object in Object Repository:

Before using this feature, ensure that the application and the specific page are open.

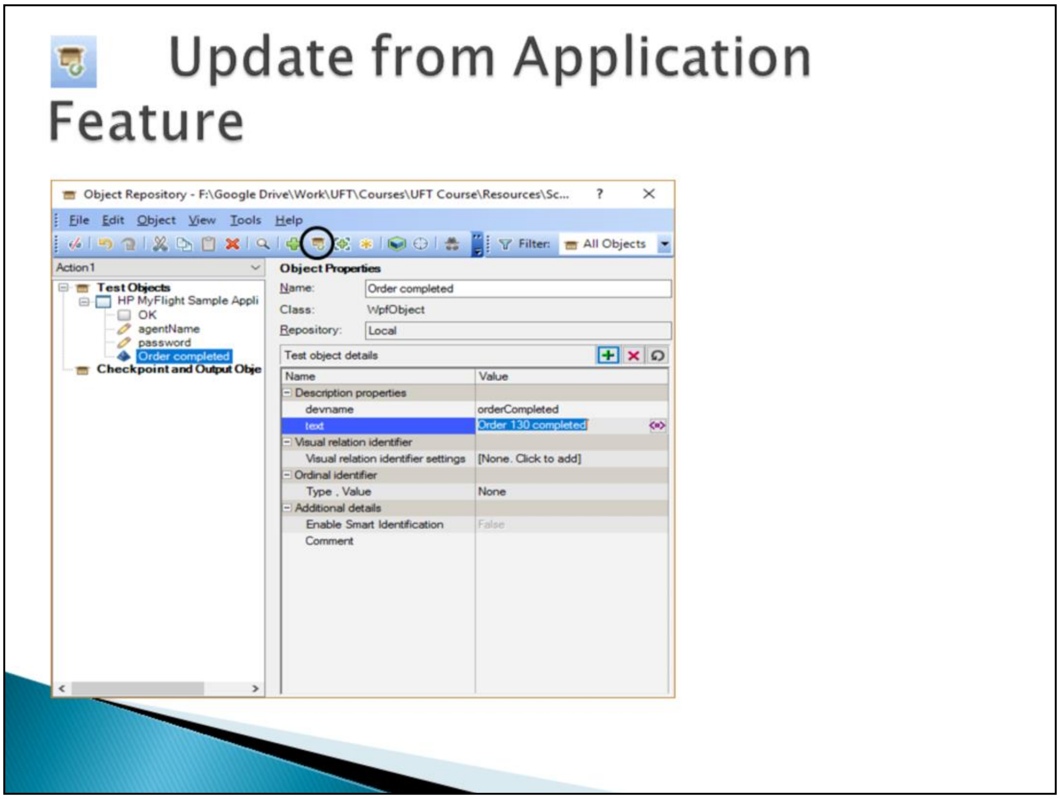
1. Open the OBJECT REPOSITORY dialog box.

2. On the toolbar, click the **LOCATE IN REPOSITORY** icon. The cursor changes to a pointing hand.

3. Use the pointing hand to click the required object in your application.

4. In the SELECT AN OBJECT dialog box, click **OK.**

The selected object is highlighted in the OBJECT REPOSITORY dialog box.



Update from Application Feature

Object Repository - F:\Google Drive\Work\UFT\Courses\UFT Course\Resources\Sc...
File Edit Object View Tools Help
Filter: All Objects

Action1

- Test Objects
  - HP MyFlight Sample Appli
    - OK
    - agentName
    - password
    - Order completed
- Checkpoint and Output Obje

Object Properties

Name: Order completed
Class: WpfObject
Repository: Local

Test object details

Name	Value
Description properties	
devname	orderCompleted
text	Order 130 completed
Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	False
Comment	

The UPDATE FROM APPLICATION feature enables you to update the properties of the

object as identified in the OBJECT REPOSITORY to the object definition in the AUT.

To update an object:

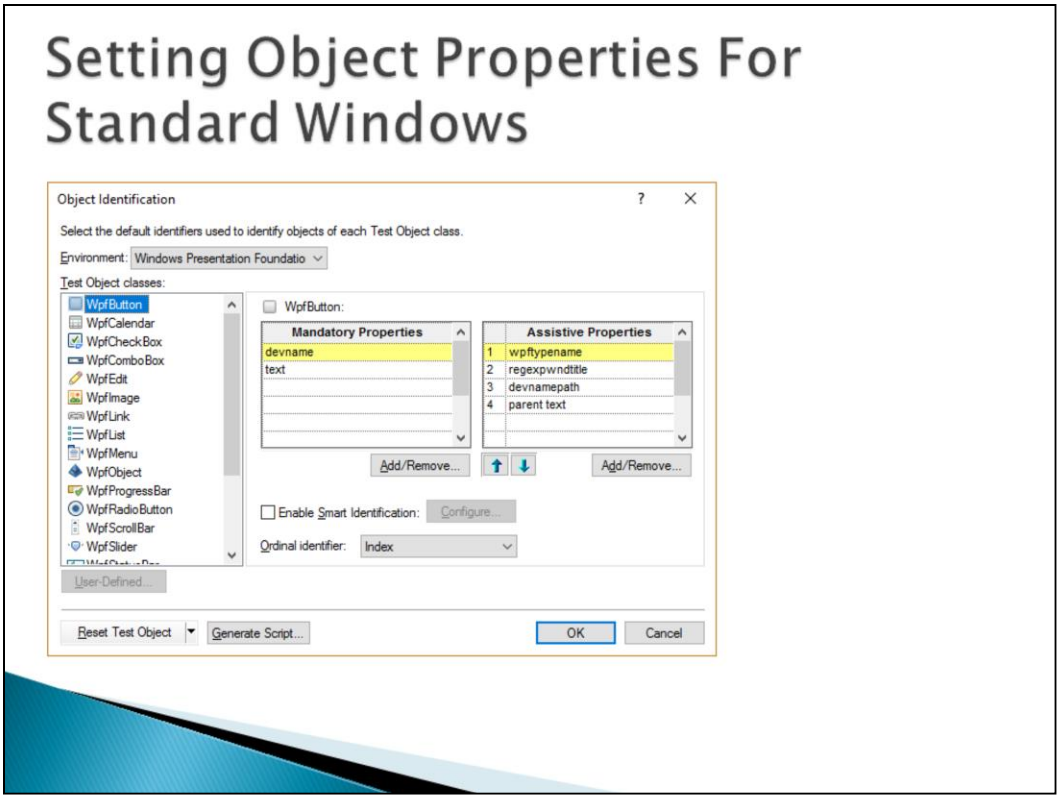
1. Open the OBJECT REPOSITORY.

2. In the left pane of the OBJECT REPOSITORY dialog box, select the test object whose description you want to update.

3. On the UFT toolbar, click the **UPDATE FROM APPLICATION** button. The UFT window is minimized and the cursor changes to a pointing hand.

4. With the pointing hand cursor, click the object in your application whose properties you want to update in the Object Repository.

5. In the SELECT AN OBJECT dialog box, click **OK.**



Setting Object Properties For Standard Windows

Object Identification

Select the default identifiers used to identify objects of each Test Object class.

Environment: Windows Presentation Foundatio

Test Object classes:

- WpfButton
- WpfCalendar
- WpfCheckBox
- WpfComboBox
- WpfEdit
- WpfImage
- WpfLink
- WpfList
- WpfMenu
- WpfObject
- WpfProgressBar
- WpfRadioButton
- WpfScrollBar
- WpfSlider

WpfButton:

Mandatory Properties
devname
text

	Assistive Properties
1	wpftypename
2	regexpwndtitle
3	devnamepath
4	parent text

□ Enable Smart Identification:

Ordinal identifier: Index

When you record a test, UFTlearns the properties and values of the objects in the test. These properties help in uniquely identifying the objects when you run the test.

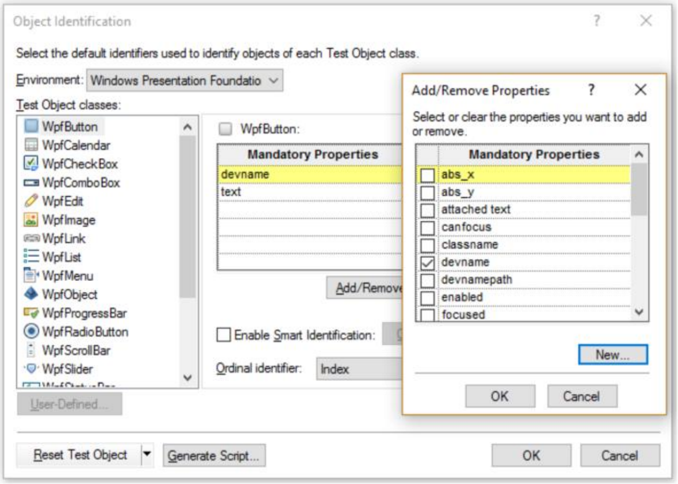
The OBJECT REPOSITORY is used to store a specific object with its properties and values for the object’s class. OBJECT IDENTIFICATION is used to define which properties to use to identify a specific object class.

In order to open OBJECT IDENTIFICATION , select **TOOLS→OBJECT IDENTIFICATION**

When you use the OBJECT IDENTIFICATION dialog box :

- View the mandatory and assistive properties of an object class.
- Modify the mandatory and assistive properties of an object class.
- Select the ordinal identifier for an object class.
- Create a user-defined object class.
- Reset the properties of an object class.

Modifying Object Class Properties



Object Identification

Select the default identifiers used to identify objects of each Test Object class.

Environment: Windows Presentation Foundatio

Test Object classes:

- WpfButton
- WpfCalendar
- WpfCheckBox
- WpfComboBox
- WpfEdit
- WpfImage
- WpfLink
- WpfList
- WpfMenu
- WpfObject
- WpfProgressBar
- WpfRadioButton
- WpfScrollBar
- WpfSlider

□ WpfButton:

Mandatory Properties
devname
text

□ Enable Smart Identification:

Ordinal identifier: Index

Add/Remove Properties

Select or clear the properties you want to add or remove.

	Mandatory Properties
□	abs_x
□	abs_y
□	attached text
□	canfocus
□	classname
☑	devname
□	devnamepath
□	enabled
□	focused

If the mandatory and assistive properties that UFTautomatically recognizes are insufficient to uniquely identify an object of an object class, you modify the properties for the object class.

For example, you want to add more properties so you select the ITEMS COUNT property to identify the objects in a test object class.

To modify the mandatory properties of an object class:

1. In the OBJECT IDENTIFICATION dialog box, below the MANDATORY PROPERTIES list, click **ADD/REMOVE**.

2. In the ADD/REMOVE PROPERTIES dialog box, clear the properties that you do not require and check the properties that you require.

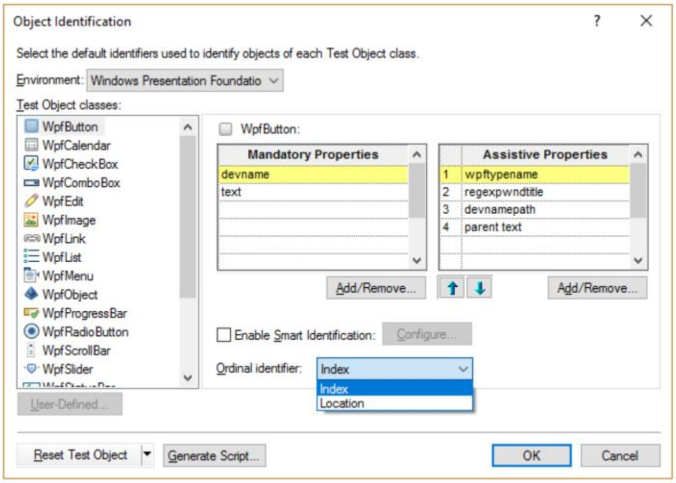
3. If the available properties are insufficient, click **NEW** to create a new property.

4. In the NEW PROPERTIES dialog box, in the NEW PROPERTY NAME field, type a name for the property, and click **OK**.

5. Click **OK** to close the ADD/REMOVE PROPERTIES dialog box and apply the new properties to the object class.

You modify the assistive properties of an object class in a similar manner.

Selecting an Ordinal Identifier



Object Identification

Select the default identifiers used to identify objects of each Test Object class.

Environment: Windows Presentation Foundatio

Test Object classes:

☐ WpfButton:

Mandatory Properties
devname
text

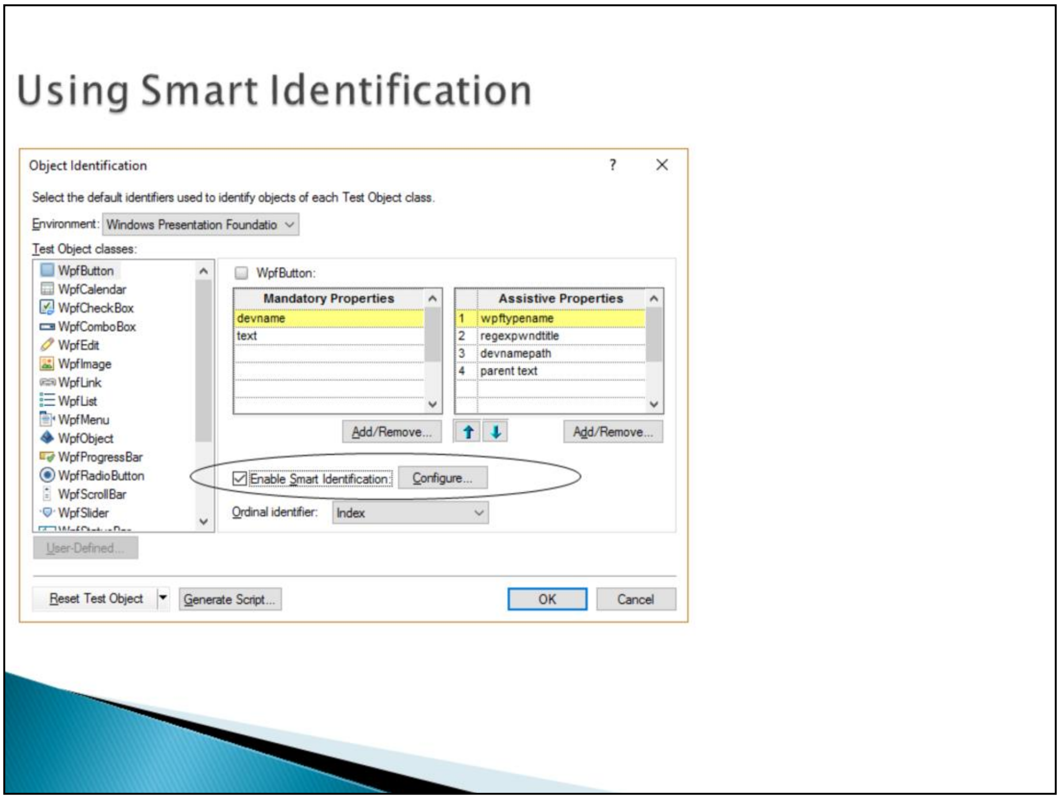
	Assistive Properties
1	wpftypename
2	regexpwndtitle
3	devnamepath
4	parent text

☐ Enable Smart Identification:

Ordinal identifier: Index

UFTuses an ordinal identifier to identify an object if the mandatory and assistive properties are insufficient to identify the object.

By default, the location of an object is used as an ordinal identifier to identify an object.



Using Smart Identification

Object Identification

Select the default identifiers used to identify objects of each Test Object class.

Environment: Windows Presentation Foundatio

Test Object classes:

- WpfButton
- WpfCalendar
- WpfCheckBox
- WpfComboBox
- WpfEdit
- WpfImage
- WpfLink
- WpfList
- WpfMenu
- WpfObject
- WpfProgressBar
- WpfRadioButton
- WpfScrollBar
- WpfSlider

□ WpfButton:

Mandatory Properties
devname
text

	Assistive Properties
1	wpftypename
2	regexpwndtitle
3	devnamepath
4	parent text

☑ Enable Smart Identification

Ordinal identifier: Index

When you run a test, UFTsearches for an object based on the description it learned for the object. If UFTcannot find any object that matches the description, or if it finds multiple objects that match, UFTuses the SMART IDENTIFICATION feature to identify the object.

The SMART IDENTIFICATION feature tries to find a unique set of properties to identify an object. This feature is normally required in tests that you create for Web-based applications

.

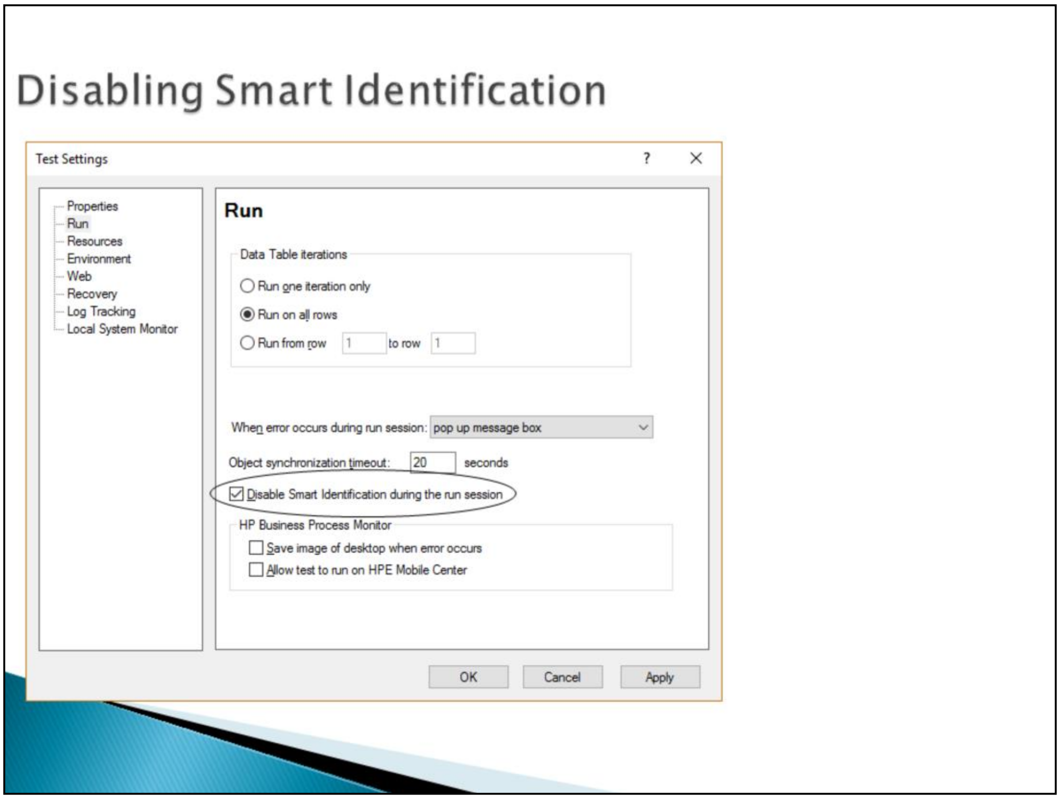
The SMART IDENTIFICATION feature is disabled by default.

To enable the SMART IDENTIFICATION feature:

1. In the OBJECT IDENTIFICATION dialog box, from the TEST OBJECT CLASSES list, select the object class for which you want to enable SMART IDENTIFICATION.

2. Check the **ENABLE SMART IDENTIFICATION** check box to enable the SMART IDENTIFICATION feature.

3. Click **OK** to close the OBJECT IDENTIFICATION dialog box.



Disabling Smart Identification

Run

Data Table iterations

○ Run one iteration only

● Run on all rows

○ Run from row 1 to row 1

When error occurs during run session: pop up message box

Object synchronization timeout: 20 seconds

☑ Disable Smart Identification during the run session

HP Business Process Monitor

□ Save image of desktop when error occurs

□ Allow test to run on HPE Mobile Center

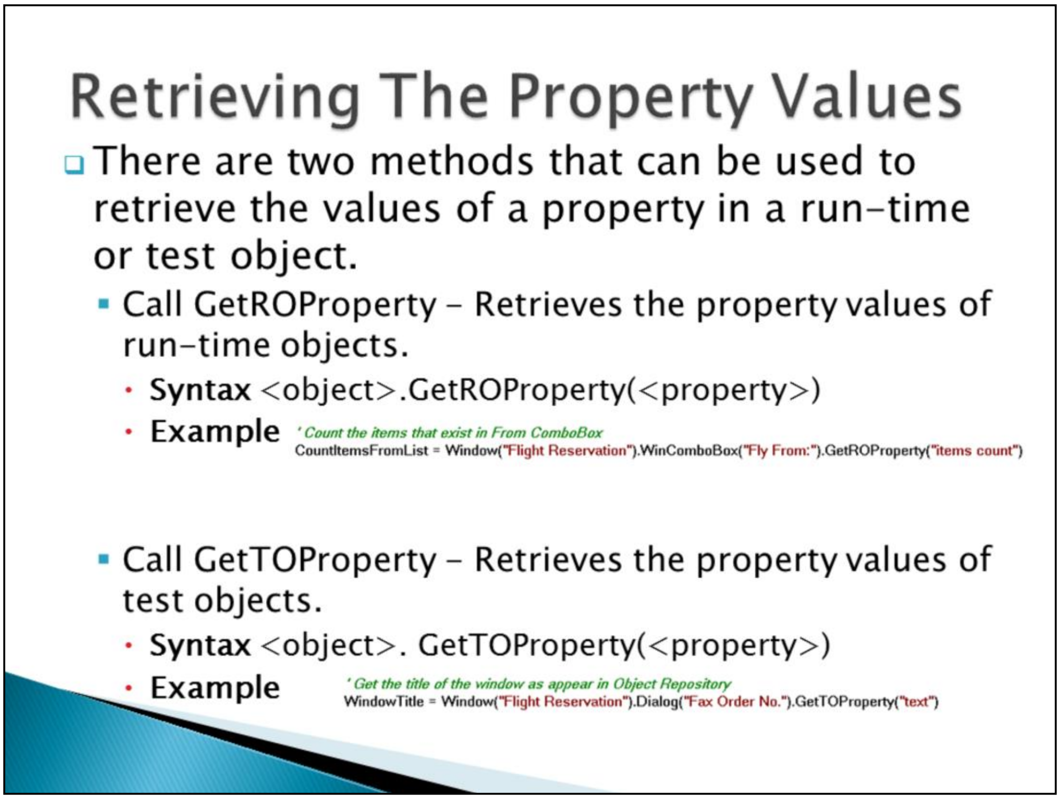
If the mandatory and assistive properties of the object classes are sufficient to uniquely identify objects during a run session, you can disable the SMART IDENTIFICATION feature during the run session.

To disable the SMART IDENTIFICATION feature:

1. From the UFT menu bar, select **FILE→SETTINGS→RUN**

2. Check the **DISABLE SMART IDENTIFICATION DURING THE RUN SESSION** check box.

3. Click **OK** to close the TEST SETTINGS dialog box.



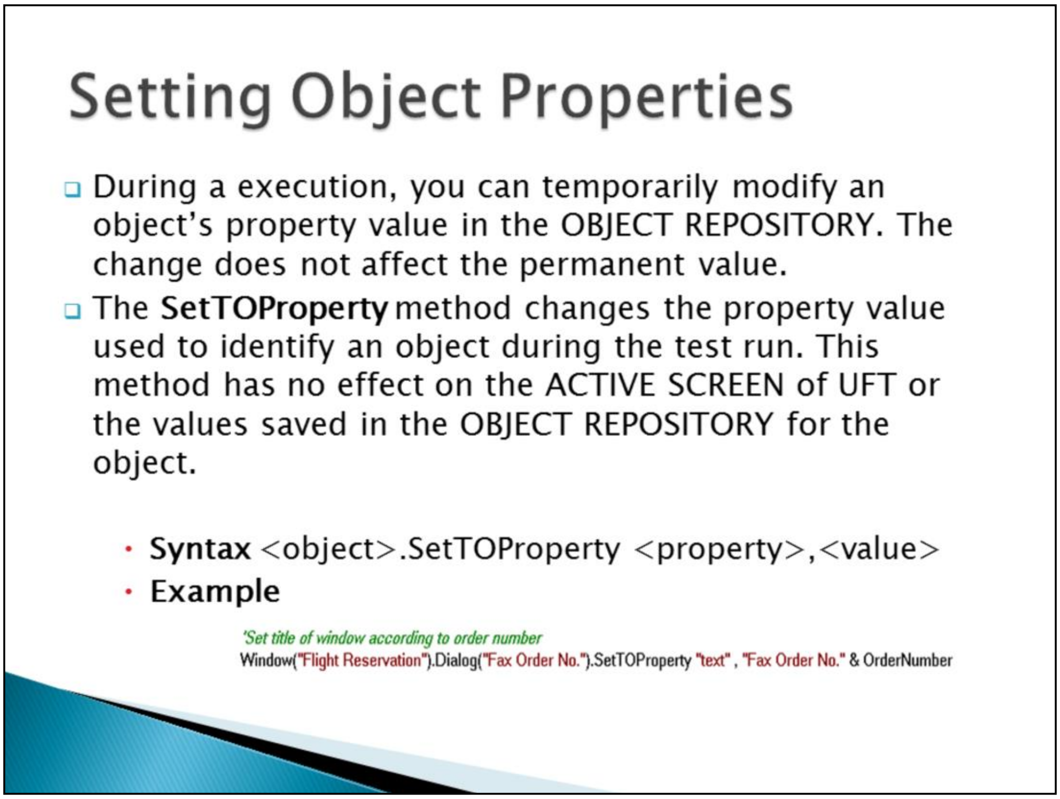
Retrieving The Property Values

- There are two methods that can be used to retrieve the values of a property in a run-time or test object.
  - Call GetROProperty - Retrieves the property values of run-time objects.
    - **Syntax** <object>.GetROProperty(<property>)
    - **Example**

```
' Count the items that exist in From ComboBox
CountItemsFromList = Window("Flight Reservation").WinComboBox("Fly From:").GetROProperty("items count")
```

  - Call GetTOProperty - Retrieves the property values of test objects.
    - **Syntax** <object>. GetTOProperty(<property>)
    - **Example**

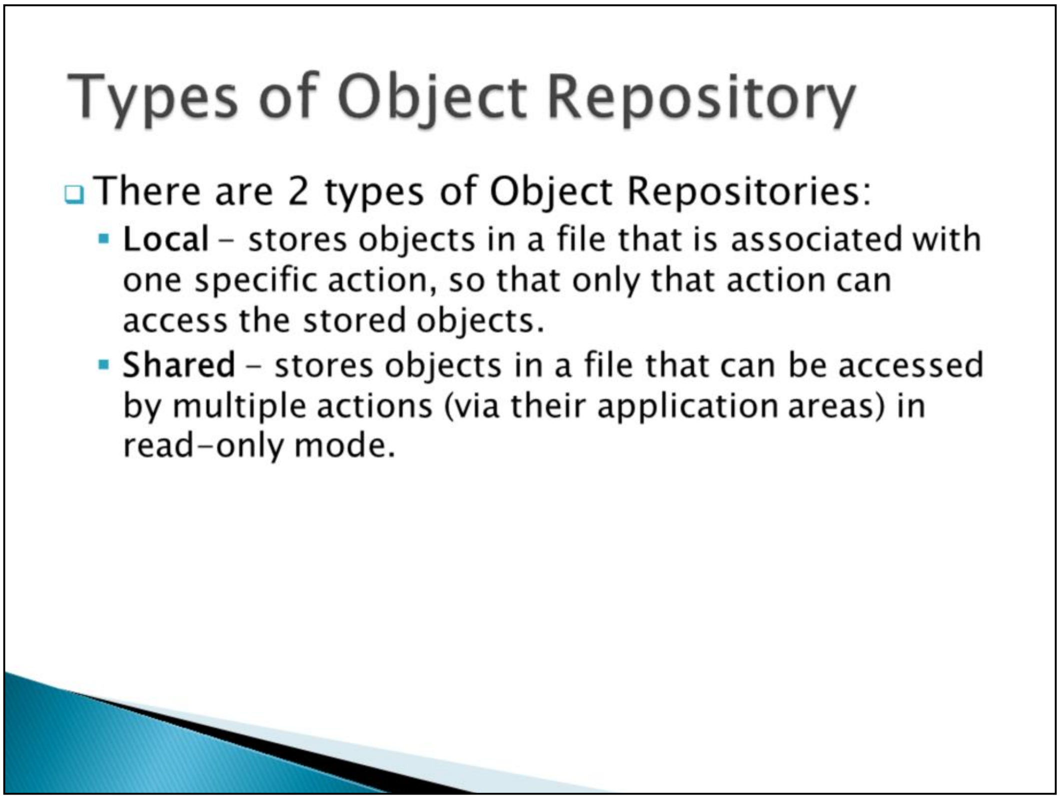
```
' Get the title of the window as appear in Object Repository
WindowTitle = Window("Flight Reservation").Dialog("Fax Order No.").GetTOProperty("text")
```



Setting Object Properties

- During a execution, you can temporarily modify an object’s property value in the OBJECT REPOSITORY. The change does not affect the permanent value.
- The **SetTOProperty** method changes the property value used to identify an object during the test run. This method has no effect on the ACTIVE SCREEN of UFT or the values saved in the OBJECT REPOSITORY for the object.
  - **Syntax** <object>.SetTOProperty <property>,<value>
  - **Example**

```
'Set title of window according to order number
Window("Flight Reservation").Dialog("Fax Order No.").SetTOProperty "text" , "Fax Order No." & OrderNumber
```



Types of Object Repository

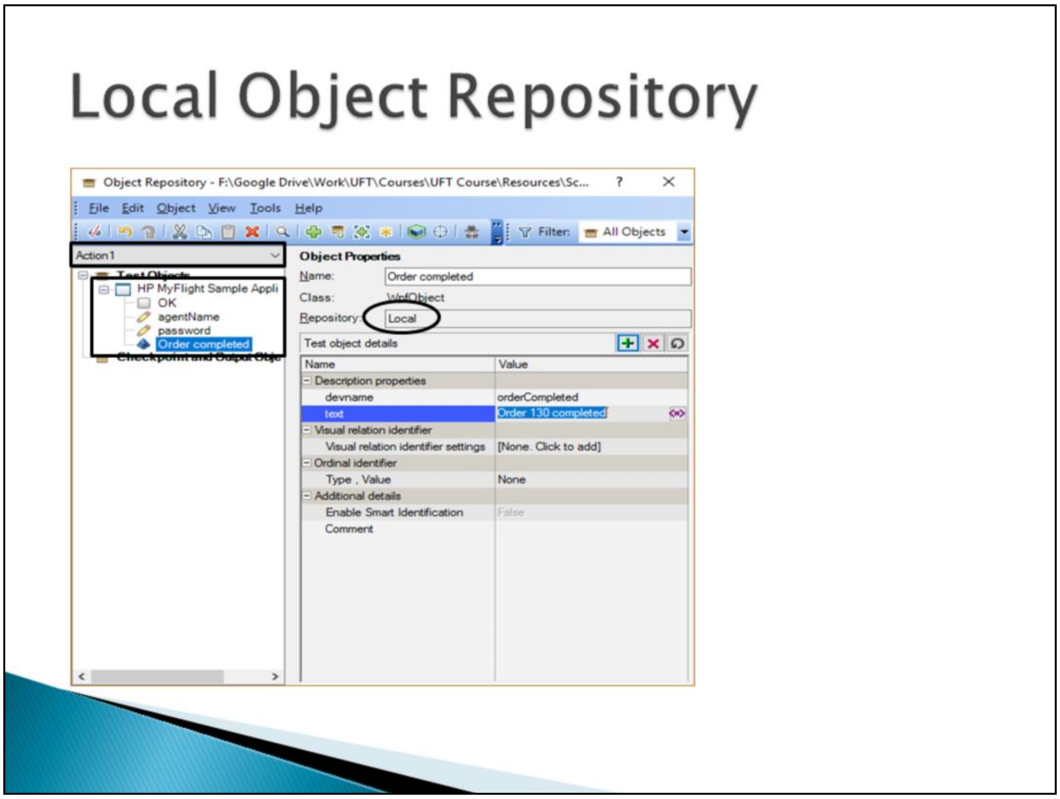
- There are 2 types of Object Repositories:
  - **Local** – stores objects in a file that is associated with one specific action, so that only that action can access the stored objects.
  - **Shared** – stores objects in a file that can be accessed by multiple actions (via their application areas) in read-only mode.

When you create an action in a test, UFT creates a new local object repository that stores objects used in the action. This local object repository is the default type of object repository.

When you associate a shared object repository with a new action, the objects in the shared object repository appear in the object repository.

When you plan and create tests, you decide how you want to store the objects in your tests. You can store the objects for each action in its corresponding local object repository. You can also store the objects in your tests in one or more shared object repositories. For each action, you can use a combination of objects from local and shared object repositories, depending on the test requirements.

You can transfer local objects to a shared object repository, if required. This transfer reduces maintenance effort and enhances the reusability of your tests because it enables you to maintain the objects in a single, shared location instead of multiple locations.



Local Object Repository

Object Repository - F:\Google Drive\Work\UFT\Courses\UFT Course\Resources\Sc...
File Edit Object View Tools Help
Filter: All Objects
Action1
Test Objects
HP MyFlight Sample Appli
OK
agentName
password
Order completed
Checkpoint and Output Obje
Object Properties
Test object details

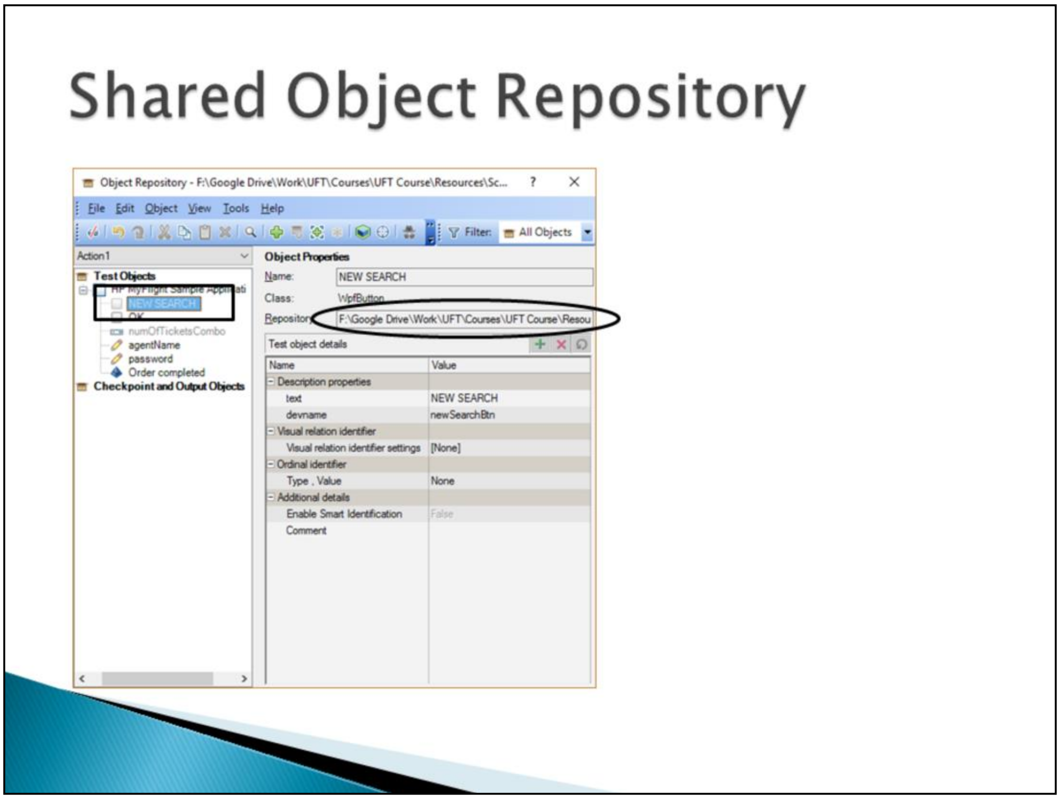
Name: Order completed

Class: WpfObject

Repository: Local

Name	Value
Description properties	
devname	orderCompleted
text	Order 130 completed
Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	False
Comment	

When you record a test, UFT automatically saves the information about the objects that you record in the local object repository. UFT creates separate local object repositories for each action.



Shared Object Repository

Object Repository - F:\Google Drive\Work\UFT\Courses\UFT Course\Resources\Sc...
File Edit Object View Tools Help
Filter: All Objects
Action1
Test Objects
HP MyFlight Sample Applicati
NEW SEARCH
OK
numOfTicketsCombo
agentName
password
Order completed
Checkpoint and Output Objects

Object Properties

Name: NEW SEARCH

Class: WpfButton

Repository: F:\Google Drive\Work\UFT\Courses\UFT Course\Resou

Test object details

Name	Value
Description properties	
text	NEW SEARCH
devname	newSearchBtn
Visual relation identifier	
Visual relation identifier settings	[None]
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	False
Comment	

If you plan to use shared objects in a test, you can create a shared object repository before you create the test. A shared object repository can be used for multiple tests.

Multiple shared object repositories can store shared objects.

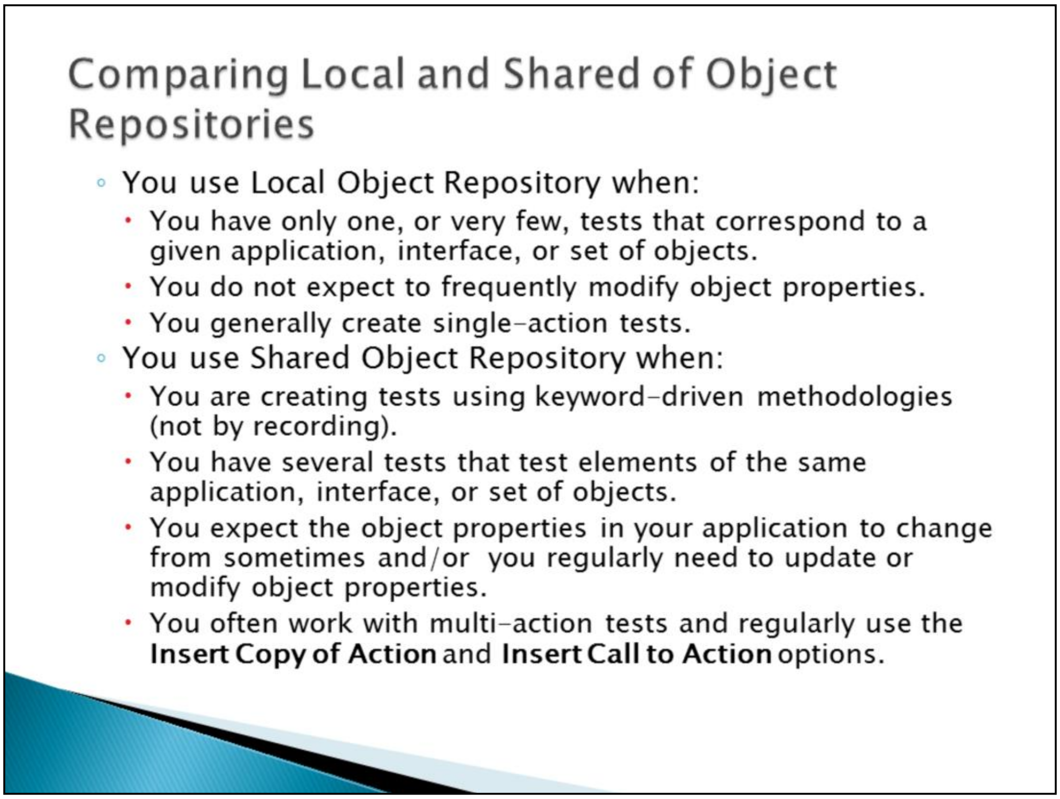
You specify the shared object repositories that you want to use for an action in a test

before you record a test. You can also create new shared object repositories and

associate these repositories with an action in a test after you record the test.

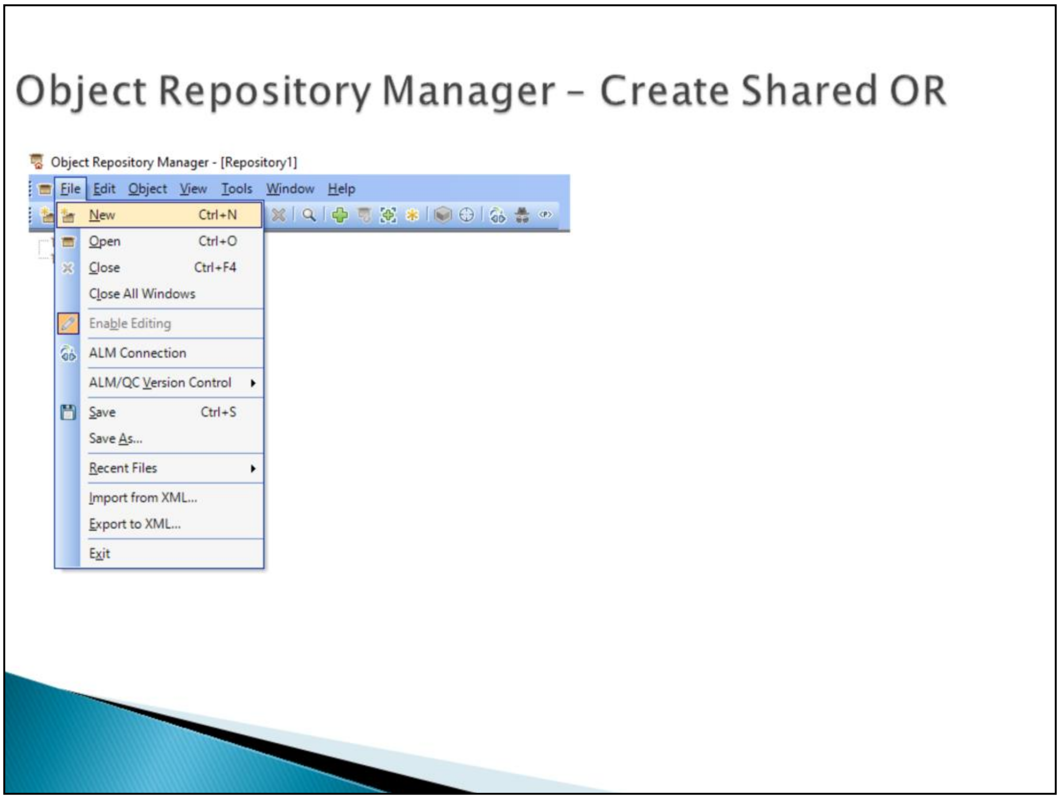
To create and manage shared object repositories, select **RESOURCES→OBJECT**

REPOSITORY MANAGER.



Comparing Local and Shared of Object Repositories

- You use Local Object Repository when:
  - You have only one, or very few, tests that correspond to a given application, interface, or set of objects.
  - You do not expect to frequently modify object properties.
  - You generally create single-action tests.
- You use Shared Object Repository when:
  - You are creating tests using keyword-driven methodologies (not by recording).
  - You have several tests that test elements of the same application, interface, or set of objects.
  - You expect the object properties in your application to change from sometimes and/or you regularly need to update or modify object properties.
  - You often work with multi-action tests and regularly use the **Insert Copy of Action** and **Insert Call to Action** options.

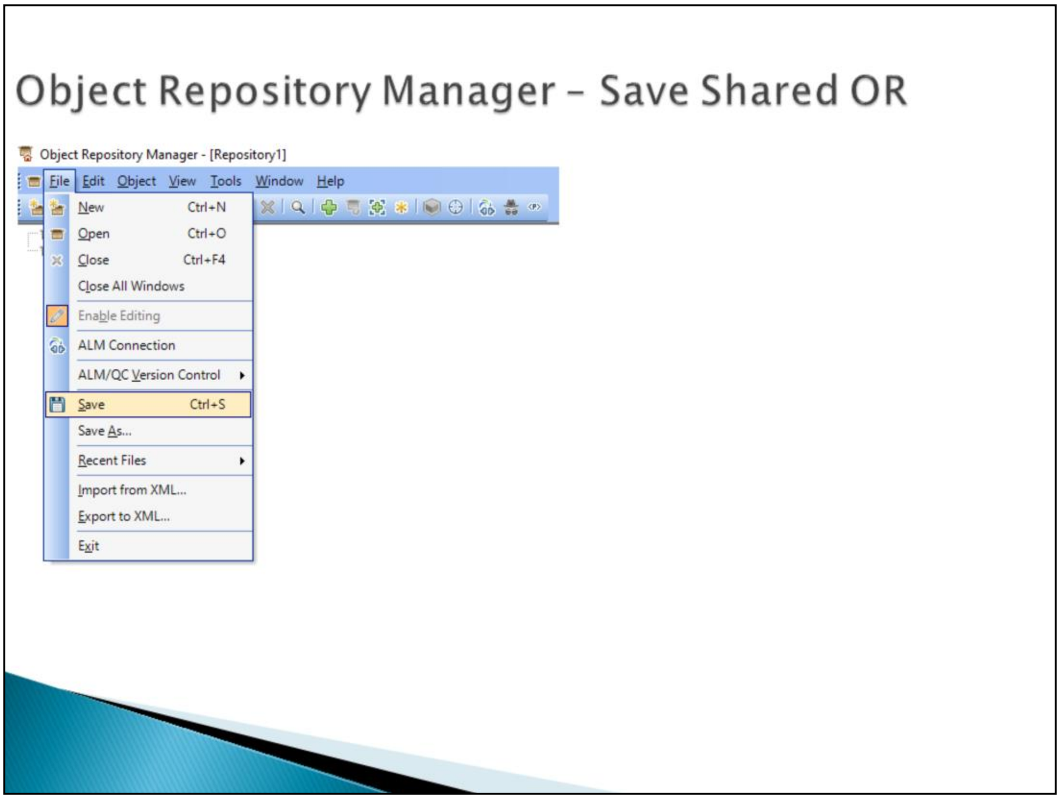


Object Repository Manager – Create Shared OR

You can create a shared object repository in the OBJECT REPOSITORY MANAGER.

To create a new shared object repository select **FILE→NEW.**

A new object repository opens.



Object Repository Manager – Save Shared OR

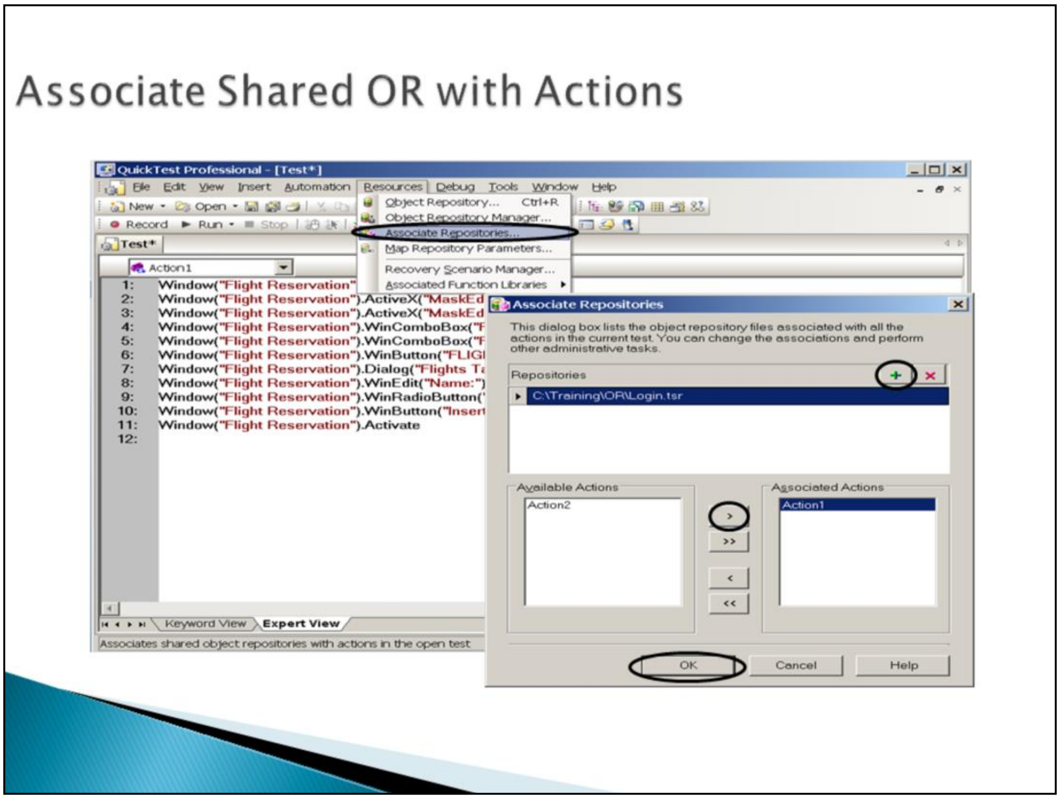
After you create or modify an object repository, you should save it. To save an object repository:

1. From the OBJECT REPOSITORY MANAGER menu bar, select **FILE→SAVE.** The SAVE SHARED OBJECT REPOSITORY dialog box appears.

2. Select the folder where you want to save the object repository.

3. In the FILE NAME field, type the name of the object repository.

4. Click **OK.** UFT saves the object repository with a .tsr extension.



Associate Shared OR with Actions

```
Window("Flight Reservation"
Window("Flight Reservation").ActiveX("MaskEd
Window("Flight Reservation").ActiveX("MaskEd
Window("Flight Reservation").WinComboBox("F
Window("Flight Reservation").WinComboBox("F
Window("Flight Reservation").WinButton("FLIGI
Window("Flight Reservation").Dialog("Flights Ta
Window("Flight Reservation").WinEdit("Name:")
Window("Flight Reservation").WinRadioButton('
Window("Flight Reservation").WinButton("Insert
Window("Flight Reservation").Activate
```

This dialog box lists the object repository files associated with all the actions in the current test. You can change the associations and perform other administrative tasks.

Repositories: C:\Training\OR\Login.tsr

Available Actions: Action2

Associated Actions: Action1

Associates shared object repositories with actions in the open test

QuickTest Professional - [Test*]
File Edit View Insert Automation Resources Debug Tools Window Help
New Open Record Run Stop
Object Repository... Ctrl+R
Object Repository Manager...
Associate Repositories...
Map Repository Parameters...
Recovery Scenario Manager...
Associated Function Libraries
Test*
Action1
Keyword View
Expert View
Associate Repositories
+
×
>
>>
<
<<
OK
Cancel
Help

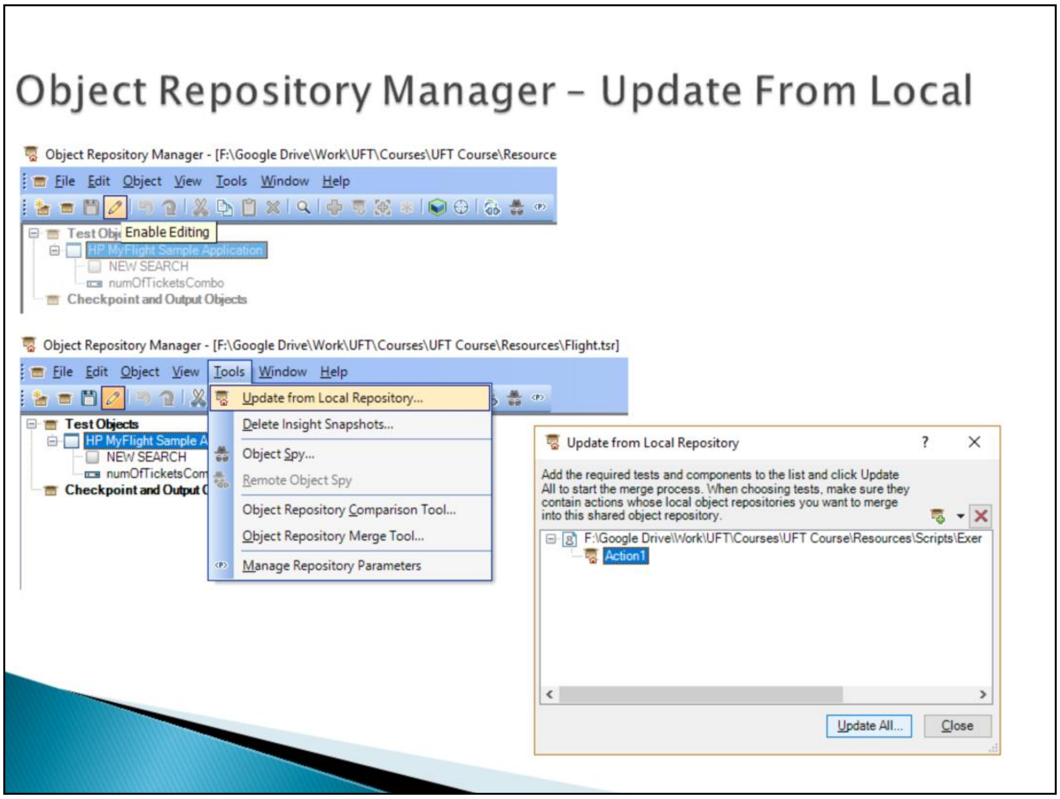
To use shared objects in a test, you first need to associate an existing shared object repository with an action. You associate the shared object repository by using the ASSOCIATE REPOSITORIES dialog box. The ASSOCIATE REPOSITORIES dialog box enables you to associate one or more shared object repositories with one or more actions in a test.

To associate a shared object repository to an action:

1. From the UFT menu bar, select **RESOURCES → ASSOCIATE REPOSITORIES. The ASSOCIATE REPOSITORIES dialog box appears.**

2. In the ASSOCIATE REPOSITORIES dialog box, click the **ADD REPOSITORY button to** select and add the repository.

3. In the AVAILABLE ACTIONS list, select the actions that you want to associate and move them to the ASSOCIATED ACTIONS list. Click **OK. The shared objects appear** in the OBJECT REPOSITORY - ALL OBJECT REPOSITORIES dialog box.



Object Repository Manager – Update From Local

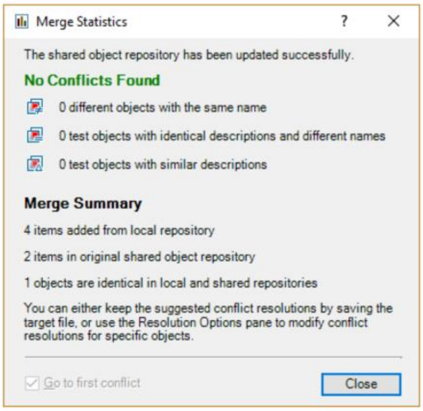
Object Repository Manager - [F:\Google Drive\Work\UFT\Courses\UFT Course\Resource
File Edit Object View Tools Window Help
Enable Editing
Test Obj
HP MyFlight Sample Application
NEW SEARCH
numOfTicketsCombo
Checkpoint and Output Objects
Object Repository Manager - [F:\Google Drive\Work\UFT\Courses\UFT Course\Resources\Flight.tsr]
File Edit Object View Tools Window Help
Update from Local Repository...
Delete Insight Snapshots...
Object Spy...
Remote Object Spy
Object Repository Comparison Tool...
Object Repository Merge Tool...
Manage Repository Parameters
Test Objects
HP MyFlight Sample A
NEW SEARCH
numOfTicketsCom
Checkpoint and Output C
Update from Local Repository
?
F:\Google Drive\Work\UFT\Courses\UFT Course\Resources\Scripts\Exer
Action1
Update All...
Close

Add the required tests and components to the list and click Update All to start the merge process. When choosing tests, make sure they contain actions whose local object repositories you want to merge into this shared object repository.

To update shared OR from local OR:

1. Associate your test to the shared OR
2. Close your test in UFT
3. Update shared OR in Object Repository Manager

Object Repository Manager – Update From Local



The shared object repository has been updated successfully.

No Conflicts Found

0 different objects with the same name

0 test objects with identical descriptions and different names

0 test objects with similar descriptions

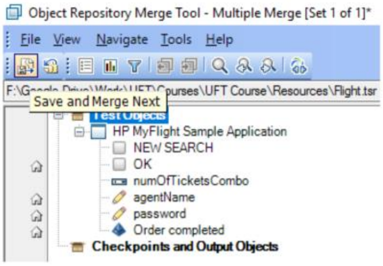
Merge Summary

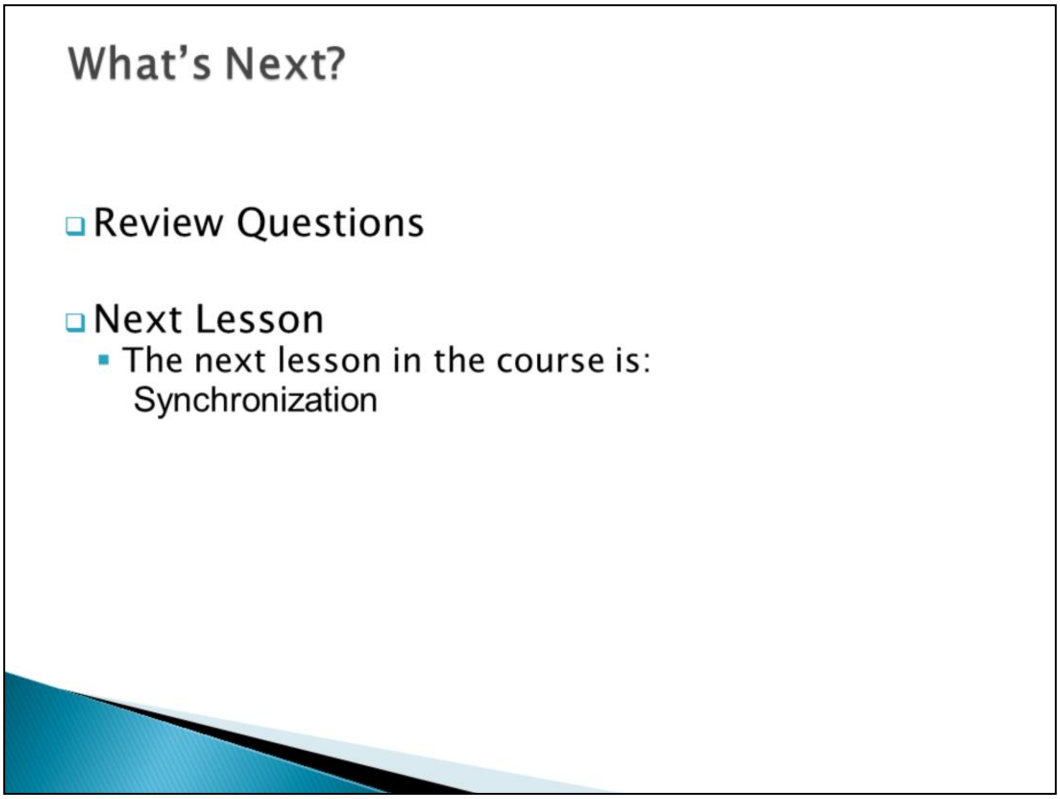
4 items added from local repository

2 items in original shared object repository

1 objects are identical in local and shared repositories

You can either keep the suggested conflict resolutions by saving the target file, or use the Resolution Options pane to modify conflict resolutions for specific objects.


Object Repository Merge Tool - Multiple Merge [Set 1 of 1]*
File
View
Navigate
Tools
Help
Save and Merge Next
F:\G ... ourses\UFT Course\Resources\Flight.tsr
Test Objects
HP MyFlight Sample Application
NEW SEARCH
OK
numOfTicketsCombo
agentName
password
Order completed
Checkpoints and Output Objects



What’s Next?

- Review Questions
- Next Lesson
  - The next lesson in the course is: Synchronization

End of Lesson