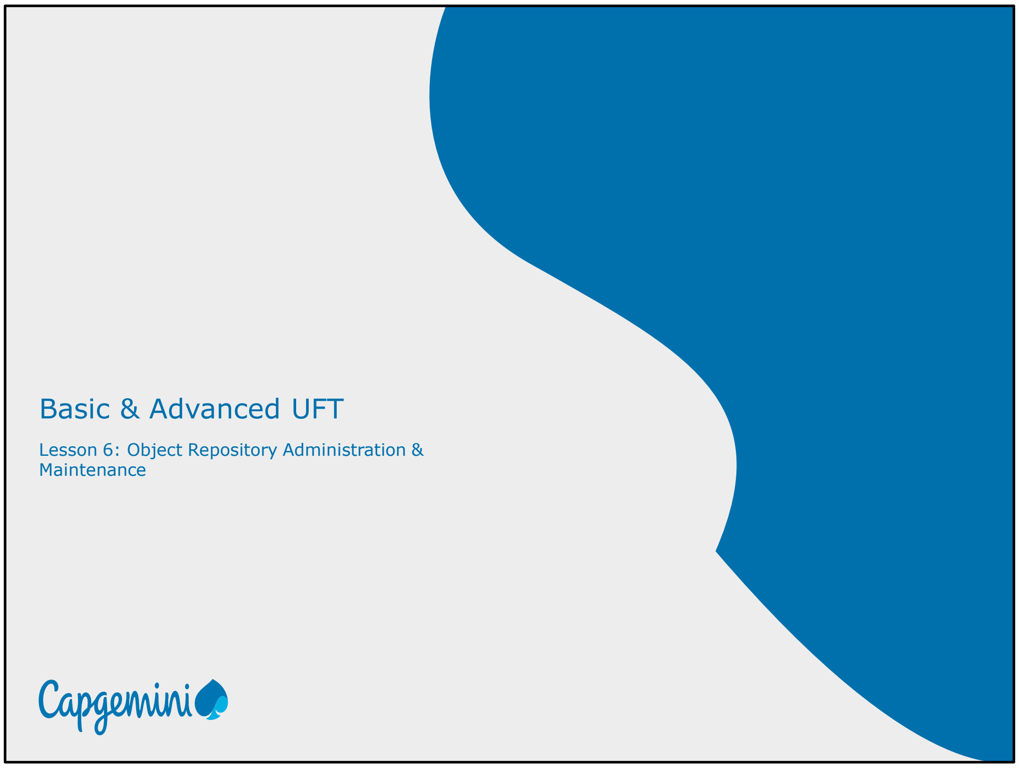


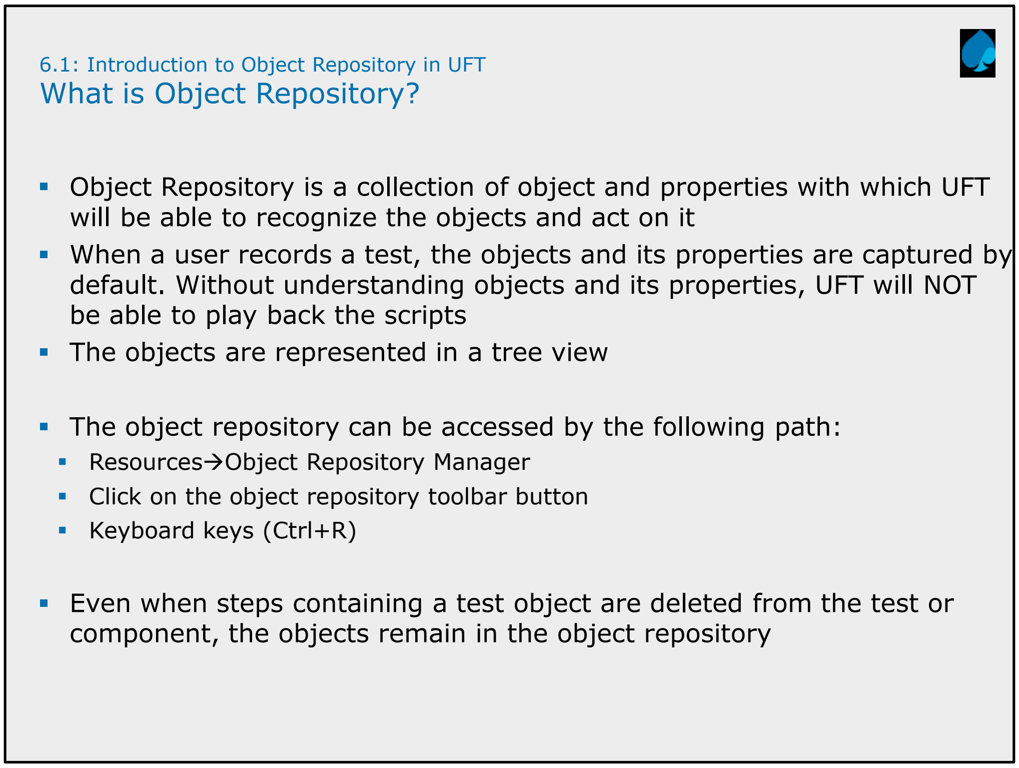
Basic & Advanced UFT

Lesson 6: Object Repository Administration & Maintenance

Capgemini

Lesson Objectives

- What is Object Repository
- Types of Object Repository
- How UFT Identifies the Object
- Test Object and Run Time Object
- Object Identification
- Ordinal Identifier
- Object repository Manager
- Managing repository Manager
- Object Repository Merge Tool
- Object Repository Comparison Tool
- Object Spy



6.1: Introduction to Object Repository in UFT

What is Object Repository?

- Object Repository is a collection of object and properties with which UFT will be able to recognize the objects and act on it
- When a user records a test, the objects and its properties are captured by default. Without understanding objects and its properties, UFT will NOT be able to play back the scripts
- The objects are represented in a tree view
- The object repository can be accessed by the following path:
  - Resources→Object Repository Manager
  - Click on the object repository toolbar button
  - Keyboard keys (Ctrl+R)
- Even when steps containing a test object are deleted from the test or component, the objects remain in the object repository

6.1: Introduction to Object Repository in UFT

Types of Object Repository

- ***Local Object Repository :*** UFT uses a separate object repository for each action. When you save your test, all of the local object repositories are automatically saved with the test (as part of each action within the test).

- ***Shared Object Repository :*** UFT uses the same object repository for different actions. You can export the local objects to a shared object repository

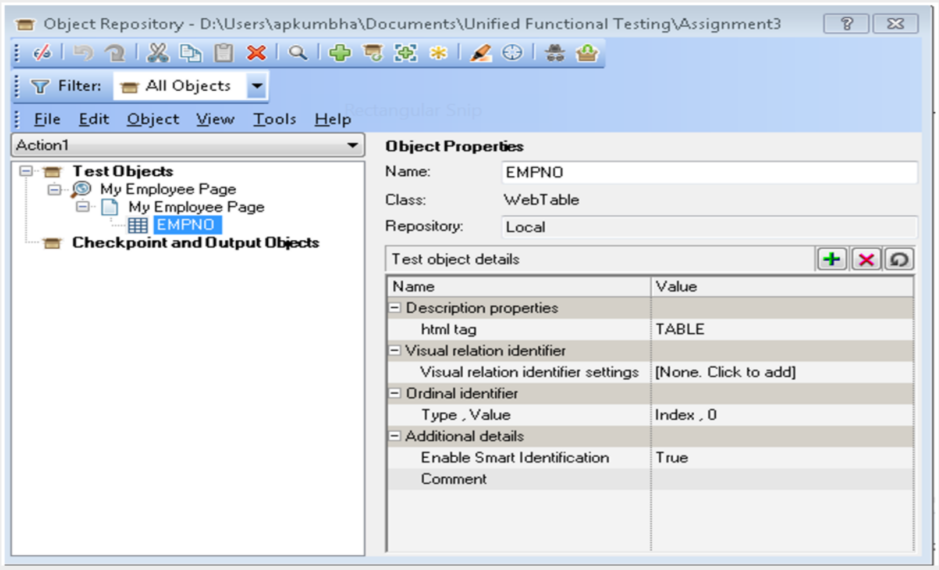
6.2: Identifying Objects in UFT

How UFT Identifies the Object

- Tests dynamically changing application by learning and identifying test objects and their expected properties and values
- Analyzes each object in a application much the same way that a person would look at a photograph and remember its details
- If the recorded description does not enable UFT to identify the specified object in a step, and a Smart Identification definition is defined (and enabled) for the object, then UFT tries to identify the object using the Smart Identification mechanism

6.2: Identifying Objects in UFT

Test Objects in Object Repository


Object Repository - D:\Users\apkumbha\Documents\Unified Functional Testing\Assignment3

Filter: All Objects

File Edit Object View Tools Help

Action1

- Test Objects
  - My Employee Page
    - My Employee Page
      - EMPNO
- Checkpoint and Output Objects

Object Properties

Name:	EMPNO
Class:	WebTable
Repository:	Local

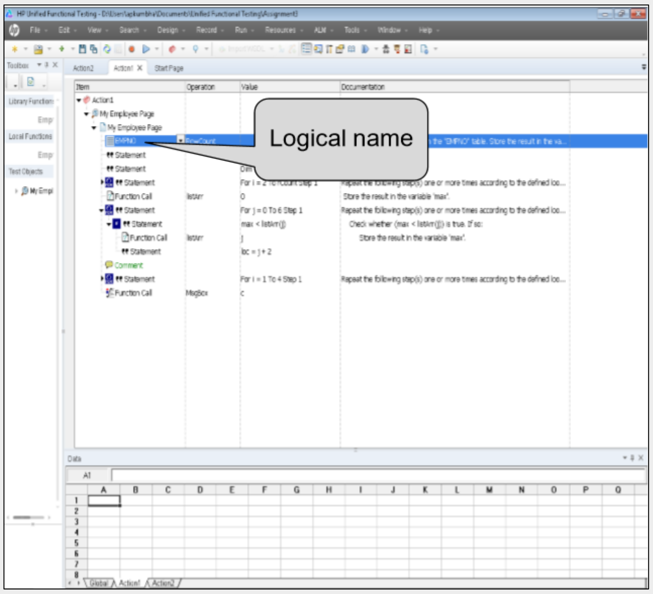
Test object details

Name	Value
Description properties	
html tag	TABLE
Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
Ordinal identifier	
Type , Value	Index , 0
Additional details	
Enable Smart Identification	True
Comment	

6.2: Identifying Objects in UFT

Logical Name of an Object

- After reading the class and properties of an object, UFT assigns a logical name to the object.
- UFT refers to the recorded objects by its logical name.
- The logical name can be edited and used in UFT scripts


HP Unified Functional Testing - D:\Users\apkumbha\Documents\Unified Functional Testing\Assignment
File Edit View Search Design Record Run Resources ALM Tools Window Help
Toolbox
Action2 Action1 Start Page
Library Functions
Local Functions
Test Objects
Logical name
Data
Global Action1 Action2

Item	Operation	Value	Documentation
Action1			
My Employee Page			
My Employee Page			
EMPNO	RowCount		the 'EMPNO' table. Store the result in the va...
Statement			
Statement		Dim	
Statement		For i = 2 To rCount Step 1	Repeat the following step(s) one or more times according to the defined loo...
Function Call	listArr	0	Store the result in the variable 'max'.
Statement		For j = 0 To 6 Step 1	Repeat the following step(s) one or more times according to the defined loo...
Statement		max < listArr(j)	Check whether (max < listArr(j)) is true. If so:
Function Call	listArr	j	Store the result in the variable 'max'.
Statement		loc = j + 2	
Comment			
Statement		For i = 1 To 4 Step 1	Repeat the following step(s) one or more times according to the defined loo...
Function Call	MsgBox	c	

6.2: Identifying Objects in UFT

How UFT Identifies the Object (Cont.)

- For each object class, UFT has a default set of properties that it always learns :

1. Mandatory Properties
2. Assistive properties
3. Ordinal Identifier

- Usually, only a few properties are needed to uniquely identify an object. There are 3 other ways by which UFT recognizes objects :

1. Smart Identification
2. Visual Relation Identifier
3. Insight for GUI Testing

6.2: Identifying Objects in UFT

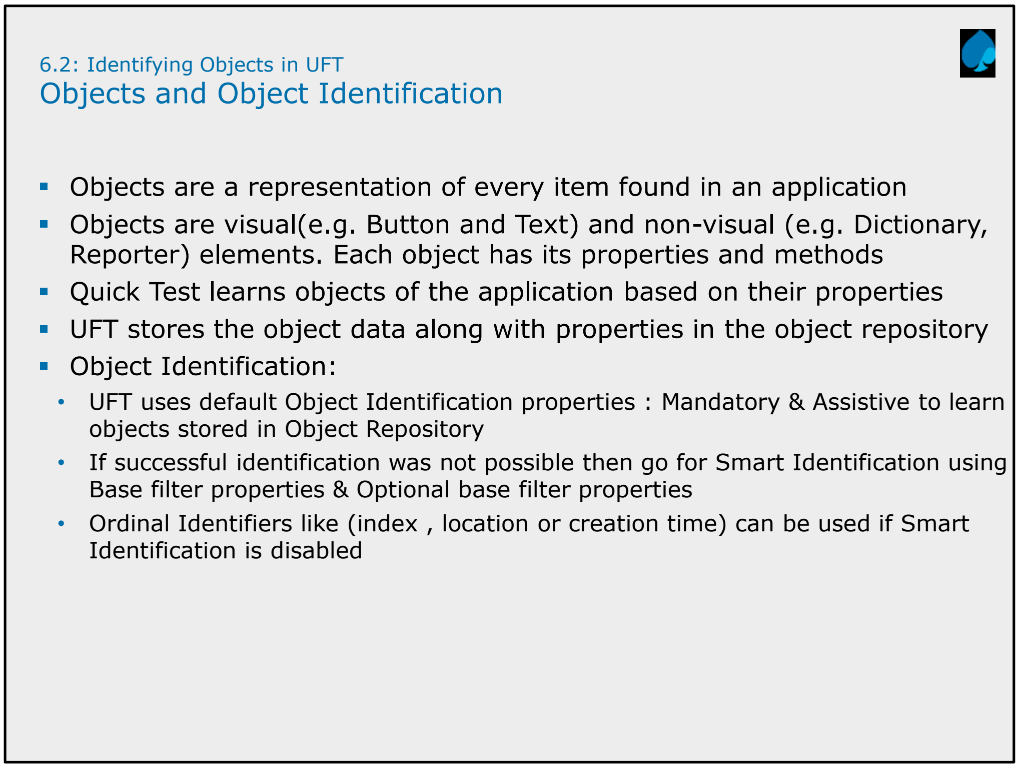
Test Object and Run Time Object

- The Test Object Model is a large set of object types or classes that it uses to represent the objects in your application
- A Test object is an object that is created in the test to represent the actual object in an application
- UFT stores information about the object that will help it identify and check the object during the test run
- A Run-time object is the actual object in a Web site or application on which methods are performed during the test run

6.2: Identifying Objects in UFT

Object Identification

- When an operation is performed on an application
- while recording a test, UFT:
  - Identifies the UFT test object class that represents the object on which UFT performed the operation and creates the appropriate test object
  - Reads the current value of the object's properties in the application and stores the list of properties and values with the test object



6.2: Identifying Objects in UFT

Objects and Object Identification

- Objects are a representation of every item found in an application
- Objects are visual(e.g. Button and Text) and non-visual (e.g. Dictionary, Reporter) elements. Each object has its properties and methods
- Quick Test learns objects of the application based on their properties
- UFT stores the object data along with properties in the object repository
- Object Identification:
  - UFT uses default Object Identification properties : Mandatory & Assistive to learn objects stored in Object Repository
  - If successful identification was not possible then go for Smart Identification using Base filter properties & Optional base filter properties
  - Ordinal Identifiers like (index , location or creation time) can be used if Smart Identification is disabled

6.2: Identifying Objects in UFT

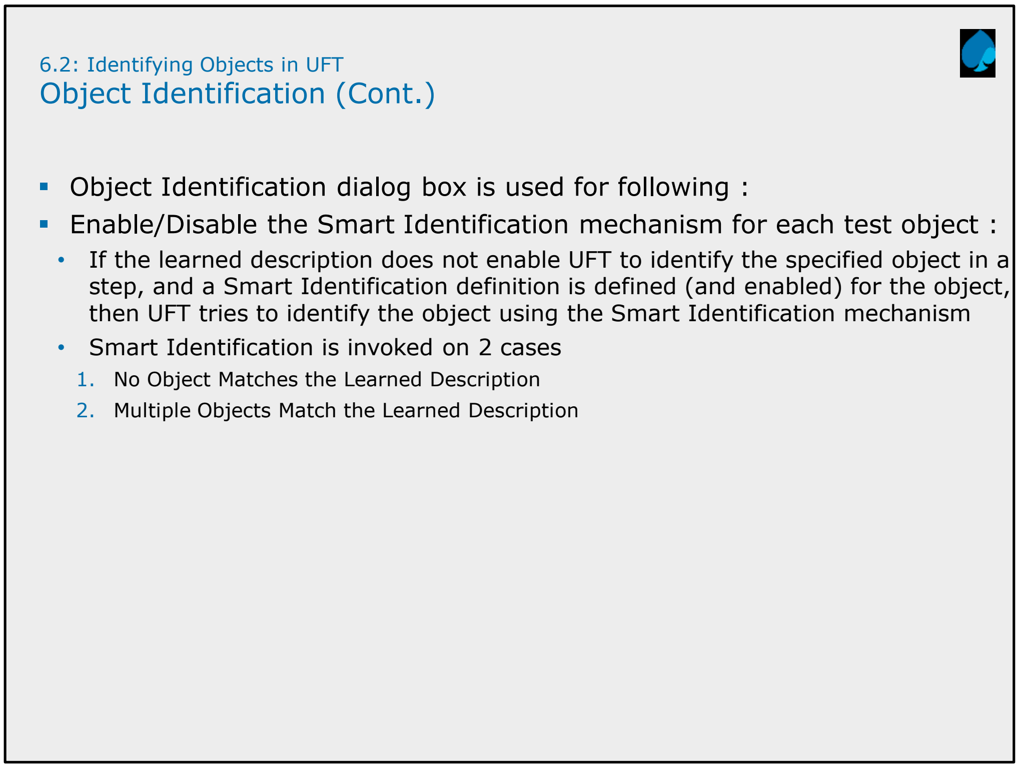
Objects and Object Identification (Cont.)

- UFT also uses Visual Relation Identifier(VRI) helps to identify application objects based on other objects that are always nearby
- Insight for GUI testing is an additional feature of UFT
- It is done by using image based identification
- UFT recognize controls in an application based on what the control looks like
- We can also use image based testing to test applications that run on non windows operating system

6.2: Identifying Objects in UFT

Object Identification

- Object Identification dialog box is used for following :
- Set/Configure mandatory and assistive properties for Test Objects :
  - If you expect that the values of the properties currently used in the object description may change, you can modify the mandatory and assistive properties that UFT learns when it learns an object of a given class
- Select the ordinal identifier for Test Objects :
  - The ordinal identifier assigns the object a numerical value that indicates its order relative to other objects with an otherwise identical description
  - This ordered value enables UFT to create a unique description when the mandatory and assistive properties are not sufficient to do so



6.2: Identifying Objects in UFT

Object Identification (Cont.)

- Object Identification dialog box is used for following :
- Enable/Disable the Smart Identification mechanism for each test object :
  - If the learned description does not enable UFT to identify the specified object in a step, and a Smart Identification definition is defined (and enabled) for the object, then UFT tries to identify the object using the Smart Identification mechanism
  - Smart Identification is invoked on 2 cases
    1. No Object Matches the Learned Description
    2. Multiple Objects Match the Learned Description

6.2: Identifying Objects in UFT

Object Identification (Cont.)

- Object Identification dialog box is used for following
  - Define user-defined object classes and map them to Standard Windows object classes :
  - The Object Mapping dialog box enables you to map an object of an unidentified or custom class to a Standard Windows class
  - You should map an object that cannot be identified only to a Standard Windows class with comparable behavior. For example, do not map an object that behaves like a button to the edit class

6.2: Identifying Objects in UFT

Ordinal Identifier

- UFT can use the following types of ordinal identifiers to identify an object:
- ***Index***: Indicates the order in which the object appears in the application code relative to other objects with an otherwise identical description
- ***Location***: Indicates the order in which the object appears within the parent window, frame, or dialog box relative to other objects with an otherwise identical description
- ***CreationTime***: (Browser object only.) Indicates the order in which the browser was opened relative to other open browsers with an otherwise identical description

6.2: Identifying Objects in UFT

Object Repository Manager

- The Object Repository Manager enables you to open multiple shared object repositories and modify them as needed
- You can open shared object repositories both from the file system and from a Quality Center project

6.2: Identifying Objects in UFT

Object Repository Manager (Cont.)

- The Object Repository Manager enables you to perform the following operations:

1. Creating New Object Repositories
2. Opening Object Repositories
3. Saving Object Repositories
4. Closing Object Repositories
5. Managing Objects in Shared Object Repositories
6. Managing Repository Parameters
7. Modifying Object Details
8. Locating Test Objects
9. Performing Merge Operations
10. Performing Import and Export Operations

6.2: Identifying Objects in UFT

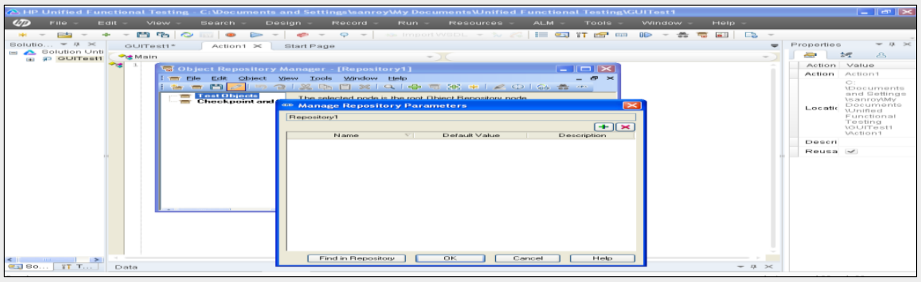
Managing Repository Parameters

- Repository parameters enable you to specify the certain property values should be parameterized, but leave the actual parameterization to be defined in each test that is associated with the object repository that contains the parameterized identification property values
- Repository parameters are useful when you want to create and run tests on an object that changes dynamically. An object may change dynamically if it is frequently updated in the application, or if its property values are set using dynamic content, for example, from a database.
- You define all the repository parameters for a specific object repository using the Manage Repository Parameters dialog box. You define each repository parameter together with an optional default value and meaningful description

6.2: Identifying Objects in UFT

Managing Repository Parameters (Cont.)

- When you open a test that uses an object repository with a repository parameter that has no default value, an indication that there is a repository parameter that needs mapping is displayed in the Missing Resources pane
- You can then map the repository parameter as needed in the test. You can also map repository parameters that have default values, and change mappings for repository parameters that are already mapped



6.2: Identifying Objects in UFT

Object Repository Merge tool

- The Object Repository Merge tool is used to merge two shared object repositories into a single shared object repository
- It is also used to merge objects from the local object repository of one or more actions into a shared object repository
- Once after merging you can view the merge statistics
- Merge Statistics describes how the files were merged, and the number and type of any conflicts that were resolved during the merge

6.2: Identifying Objects in UFT

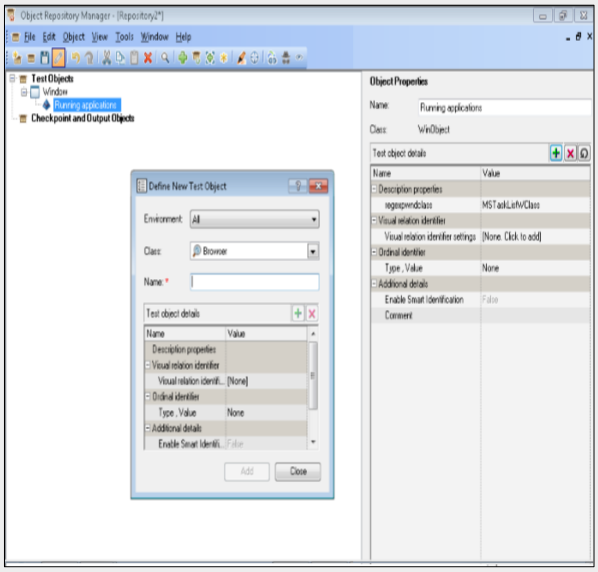
Object Repository Comparison Tool

- UFT Professional enables you to compare two shared object repositories using the Object Repository Comparison Tool, and view the differences in their objects, such as different object names, different object descriptions, and so on
- After the compare process, the Comparison Tool provides a graphic presentation of the objects in the object repositories, which are shown as nodes in a hierarchy
- Objects that have differences, as well as unique objects that are included in one object repository only, can be identified according to a color configuration that you can select
- Objects that are included in one object repository only are identified in the other object repository by the text "Does not exist". You can also view the properties and values of each object that you select in either object repository

6.2: Identifying Objects in UFT

Object Repository Manager – Define new Test Object

- Define New Test Object adds a new Test object.
- This enables to prepare an object repository and build tests or components for application before the application is ready for testing.
- Objects can be added to Local or Shared Object Repository.


Object Repository Manager - [Repository2*]
File Edit Object View Tools Window Help
Test Objects
Window
Running applications
Checkpoint and Output Objects
Define New Test Object
Add
Close
Object Properties

Define New Test Object

Environment: All

Class: Browser

Name: *

Test object details

Name	Value
Description properties	
Visual relation identifier	
Visual relation identi...	[None]
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identi...	False

Object Properties

Name: Running applications

Class: WinObject

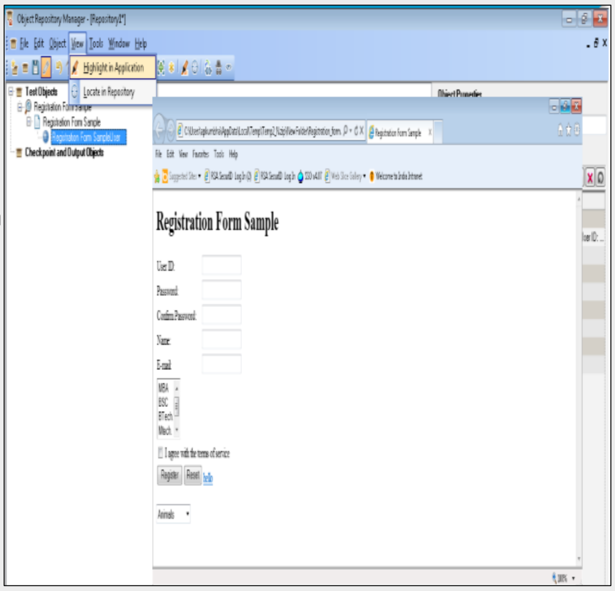
Test object details

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

6.2: Identifying Objects in UFT

Object Repository Manager – Highlight in Application

- Select a test object in object repository and highlight it in the application.
- When highlight in Application is selected, UFT indicates the selected object's location in the application by temporarily showing a frame around the object and causing it to flash briefly.
- The application must be open to the correct context so that the object is visible



6.2: Identifying Objects in UFT

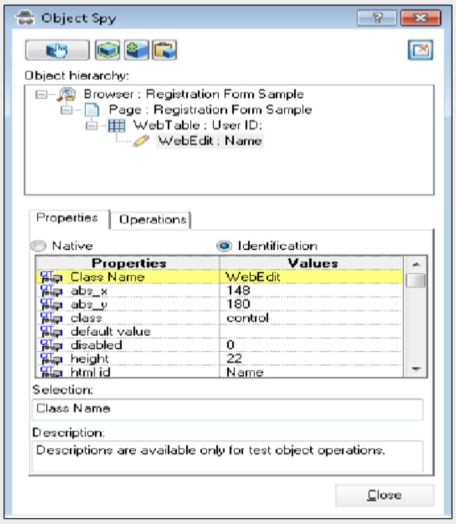
Object Spy

- Using the Object Spy, the run-time or test object properties and methods of any object in an open application can be viewed
- Use the Object Spy pointer to point to an object
- It displays the run-time or test object properties and values of the selected object in the Properties tab

6.2: Identifying Objects in UFT

Object Repository Manager – Object spy

- It enables to view the native properties and operations of any object in an open application, as well as the test object hierarchy, identification properties, and operations of the test object that UFT uses to represent that object.
- This helps in finding the current object properties of any test object.



Browser : Registration Form Sample
Page : Registration Form Sample
WebTable : User ID:
WebEdit : Name

Native [✓ Identification]

Properties	Values
Class Name	WebEdit
abs_x	148
abs_y	180
class	control
default value	
disabled	0
height	22
html id	Name

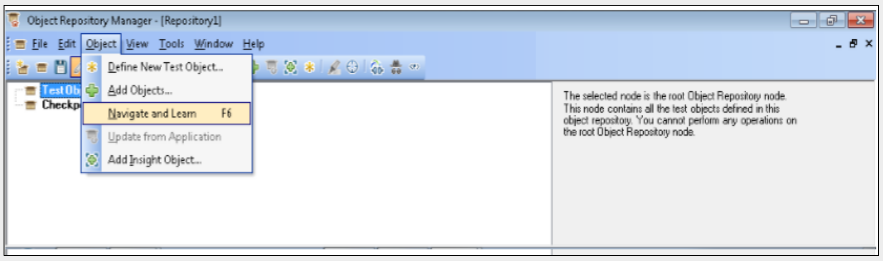
Class Name

Descriptions are available only for test object operations.

6.2: Identifying Objects in UFT

Object Repository Manager – Navigate and Learn

- The Navigate and Learn option enables to add multiple test objects to a shared object repository based on defined filter while navigating through the application

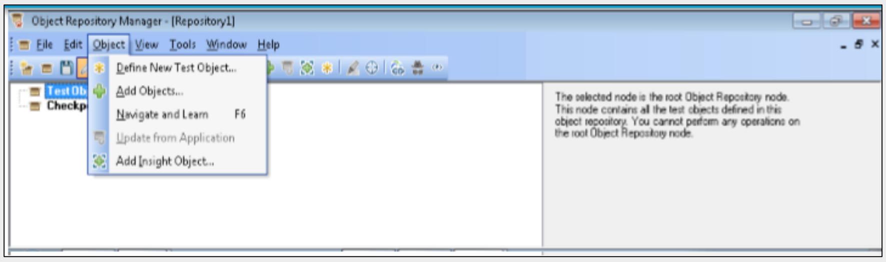

Object Repository Manager - [Repository1]
File Edit Object View Tools Window Help
Define New Test Object...
Add Objects...
Navigate and Learn F6
Update from Application
Add Insight Object...
Test Ob
Checkp

The selected node is the root Object Repository node. This node contains all the test objects defined in this object repository. You cannot perform any operations on the root Object Repository node.

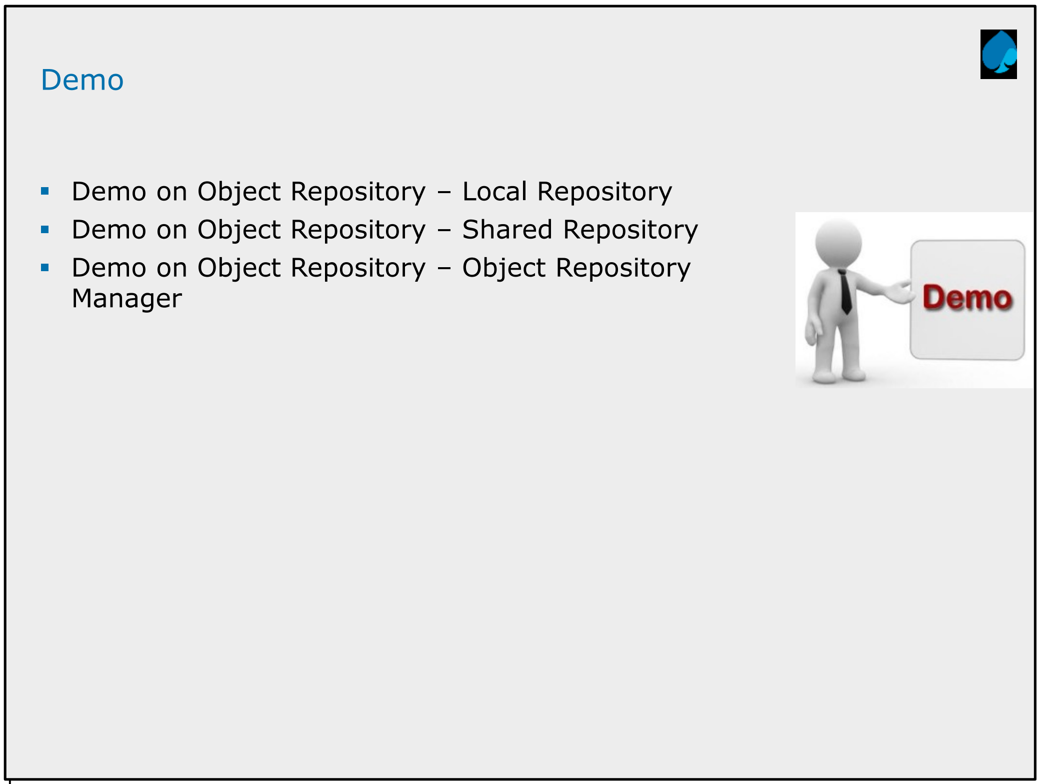
6.2: Identifying Objects in UFT

Object Repository Manager – Update from Application

- As the application changes, it is important to update individual test object properties from the object in the application using the Update from Application option.

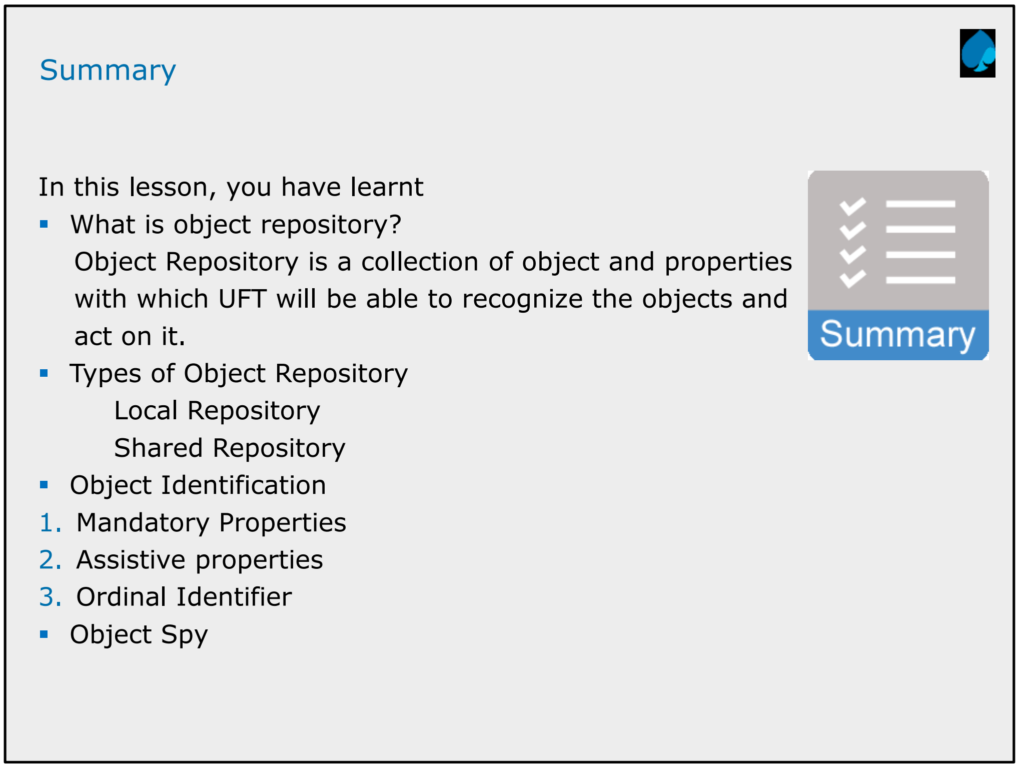


The selected node is the root Object Repository node. This node contains all the test objects defined in this object repository. You cannot perform any operations on the root Object Repository node.



Demo

- Demo on Object Repository – Local Repository
- Demo on Object Repository – Shared Repository
- Demo on Object Repository – Object Repository Manager



Summary

In this lesson, you have learnt

- What is object repository?

  Object Repository is a collection of object and properties with which UFT will be able to recognize the objects and act on it.
- Types of Object Repository
  - Local Repository
  - Shared Repository
- Object Identification

1. Mandatory Properties
2. Assistive properties
3. Ordinal Identifier

- Object Spy

Summary

Add the notes here.