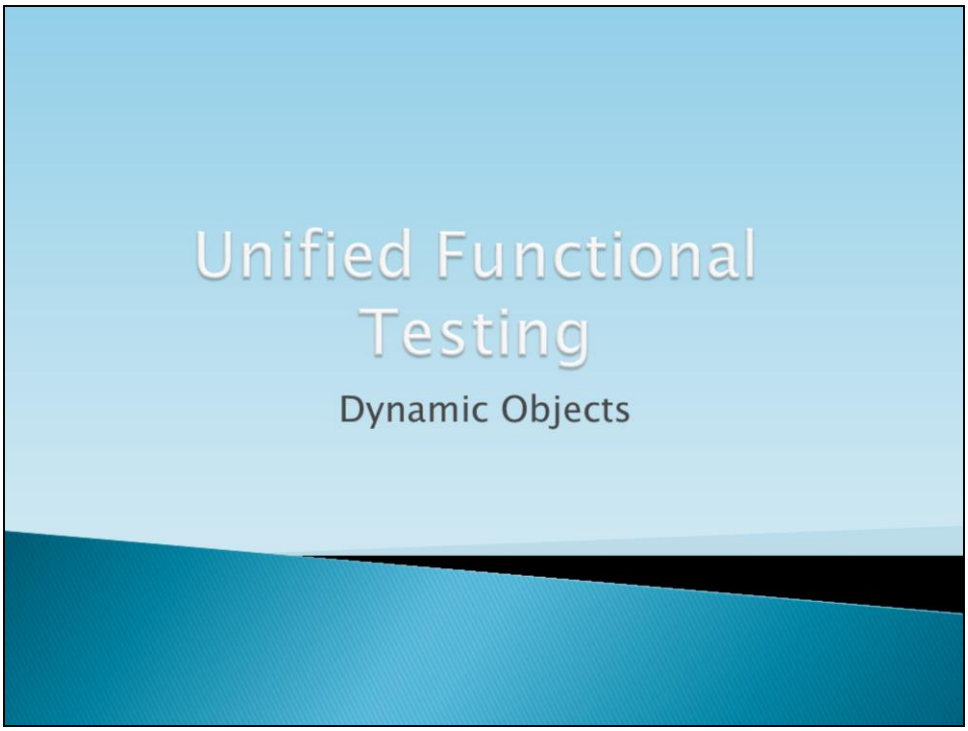

Unified Functional Testing
Dynamic Objects

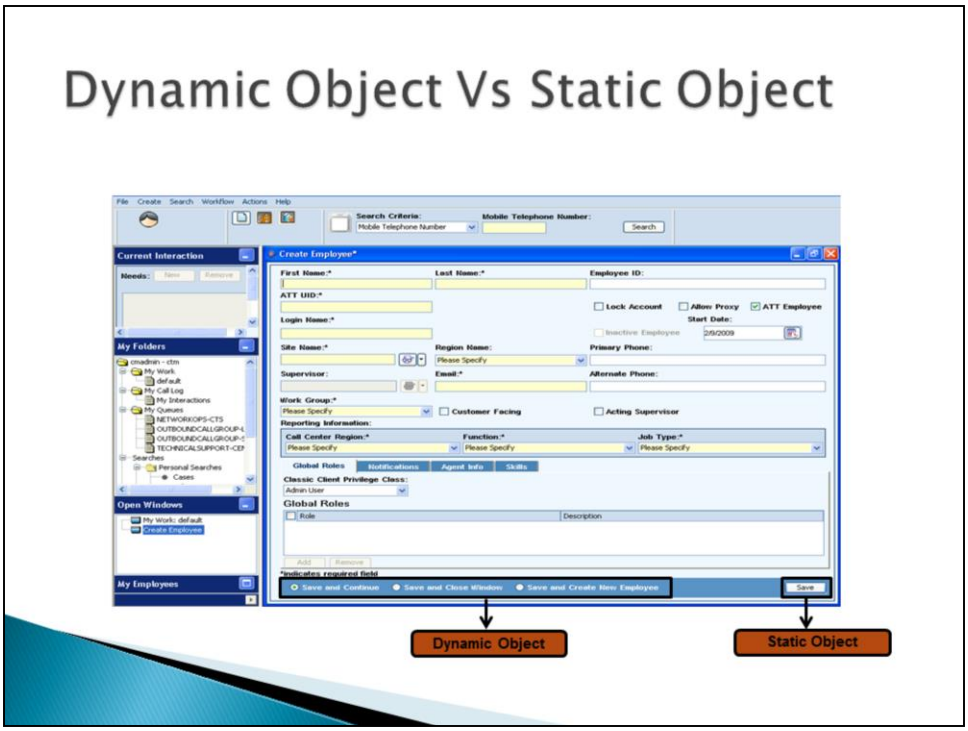
Lesson Objectives

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

- Set dynamic values of objects in the Object Repository.
- Build a Descriptive Programming for an object.
- Identify uses of descriptive programming.
- Create and use a Description object.

Topics

1. What are dynamic objects
2. Descriptive programming
3. Description object
4. SetTOProperty



A lot of applications contain static and dynamic objects.

Static Object – objects that always exist in the same state in the application.

Dynamic Object – Object that not always exist in the same state in the application because it is created during run time and according to the flow.

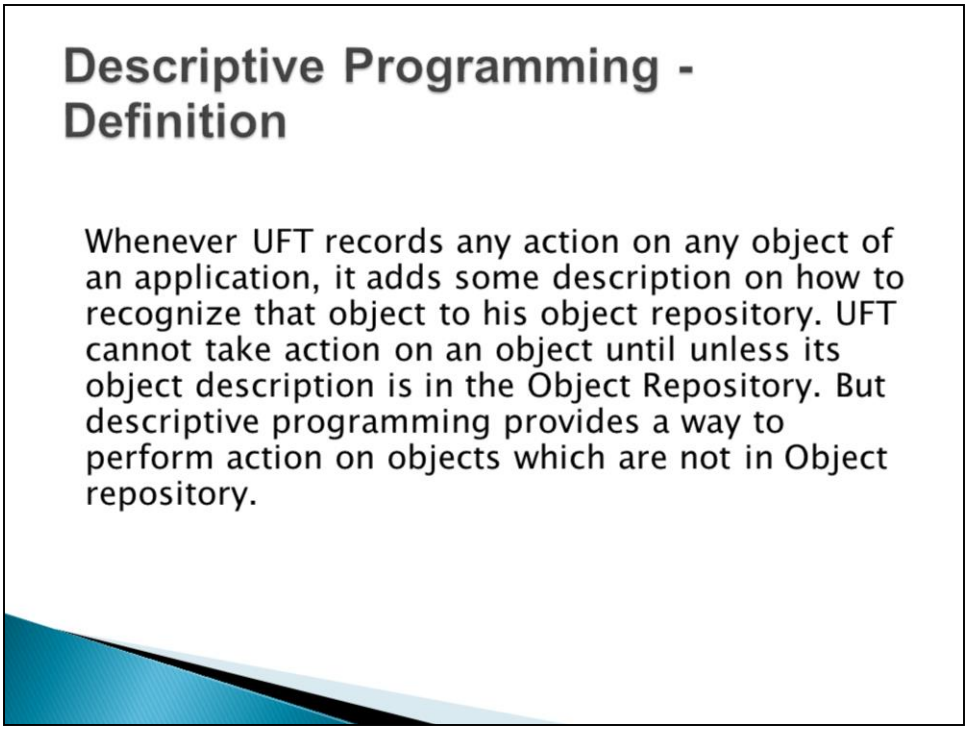
For Example , Indicates required field can change according to the workflow and Save button will be same for each workflow.

Indentify Dynamic Objects

- Two methods to handle dynamic objects in a Quick Test script:
  - Setting Dynamic values in OBJECT REPOSITORY
  - Using Descriptive Programming for dynamic objects that do not exist in the OBJECT REPOSITORY.

Types Of Setting Dynamic Values

- For Objects that exist in Object Repository:
  - Using Regular Expression.
  - Using Data from DATA TABLE.
  - Using the SetTOProperty method.
- For Objects that doesn't exist in Object Repository:
  - Using Descriptive Programming.



Below are some of the situations when Descriptive Programming can be considered useful:

1. The objects in the application are dynamic in nature and need special handling to identify the object. The best example would be of clicking a link which changes according to the user of the application.
2. When every page have the same 3 buttons, for example , “Cancel”, “Back” and “Next”.
3. Perform the same operation on a list of objects with similar properties: For example, to select all the checkboxes on a page.
4. Perform an operation on a static object that you choose based on run-time information

Identify The Object Class And Building Descriptive Programming

- Steps for Building Descriptive Programming :
  - Identify the Object Class
  - Select Object Properties
  - Create a Descriptive Programming .

Identify Object Class

- Two Options to identify the object class :
  - Identify the class of the object by locating a similar dynamic object in the OBJECT REPOSITORY.
  - Use OBJECT SPY on the desired object.

Selecting Object Properties

- After the object class is identified, select the properties that will be needed to uniquely identify the object.
- You can view object properties in the OBJECT SPY, OBJECT REPOSITORY, or in the OBJECT MODEL REFERENCE in Quick Test Professional HELP.

Ways of Creating a Descriptive Programming

- The Descriptive Programming is created using one of these methods:
  - Specifying property and value pairs in place of the logical name.
  - Using the Description object.

Specifying property and value pairs

- You can describe an object directly in a statement by specifying **property:=value** pairs describing the object instead of specifying an object's name. The general syntax is:

TestObject("PropertyName1:=PropertyValue1", "...", "PropertyNameX:=PropertyValueX")

Examples of Specifying property and value pairs

`valueX` is the value of `propertyX`. If the value contains a special regular expression character, use the backslash (\) character to instruct QuickTest to treat the special character as a literal character.

This example shows how a value is specified for the WINBUTTON window id property:


```
Window("Flight Reservation").WinButton("window id:=5").Click
```

This example shows how to use a variable and the concatenation (&) operator to specify a dynamic description for the DIALOG name property:


```
Window("Flight Reservation").Dialog("text:=Fax Order No\. " & OrderNum)
```

A regular expression may also be used in a programmatic description:


```
Window("Flight Reservation").Dialog("text:=Fax Order No\. .*")
```

Sometimes it is necessary to use more then one property and value pair to uniquely identify a dynamic object:


```
Window("Flight Reservation").Dialog("enabled:=True, visable:=True")
```

Description Object

- The Description object enables you to specify multiple properties to uniquely identify a dynamic object. To create and use the Description object:
  - 1. Create a Description object by using the following syntax:

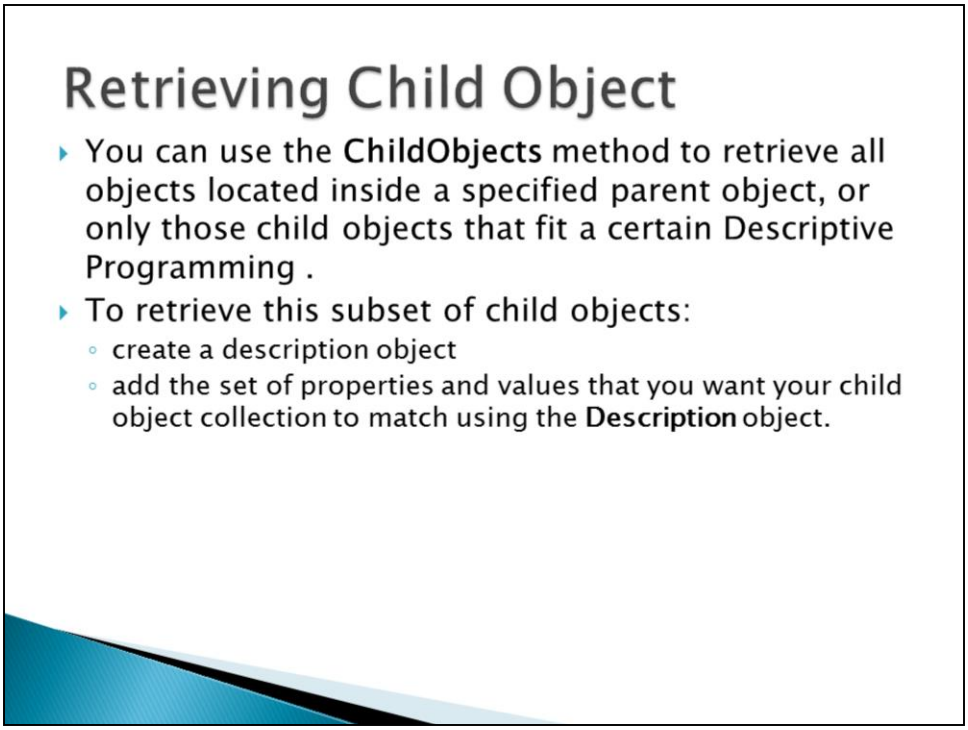
Dim objDes
Set objDes = Description.Create()

2. Set property and value pairs in the Description object by using the following syntax:

< description_object>.(<property1>).Value = <value1>
<description_object>.(<propertyX>).Value = <valueX>

3. The Description object can then be used in the script with the following syntax:

<object_hierarchy>.<object_class>(<description_object>)



Examples:

```
Set MyDescription = Description.Create()
MyDescription("html tag").Value = "INPUT"
MyDescription("type").Value = "checkbox"
Set Checkboxes =
Browser(“Amdocs").Page(“Amdocs").ChildObjects(MyDescription)
NoOfChildObjs = Checkboxes.Count
For Counter=0 to NoOfChildObjs-1
      Checkboxes(Counter).Set "ON"
Next
```

```
Set ObjDesc = Description.Create()
ObjDesc("Class Name").Value = "JavaInternalFrame"
```

```
ObjDesc("title").Value = ".*"
Set ObjF = JavaWindow("Clarify").ChildObjects(ObjDesc)
sLabel = CStr(ObjF(0).GetROProperty("label"))
ObjF(0).SetTOProperty "label",sLabel
```

What's Next?

- Review Questions
- Exercise
- Next Lesson
  - The next lesson in the course is:
    DB CheckPoints

End of Lesson