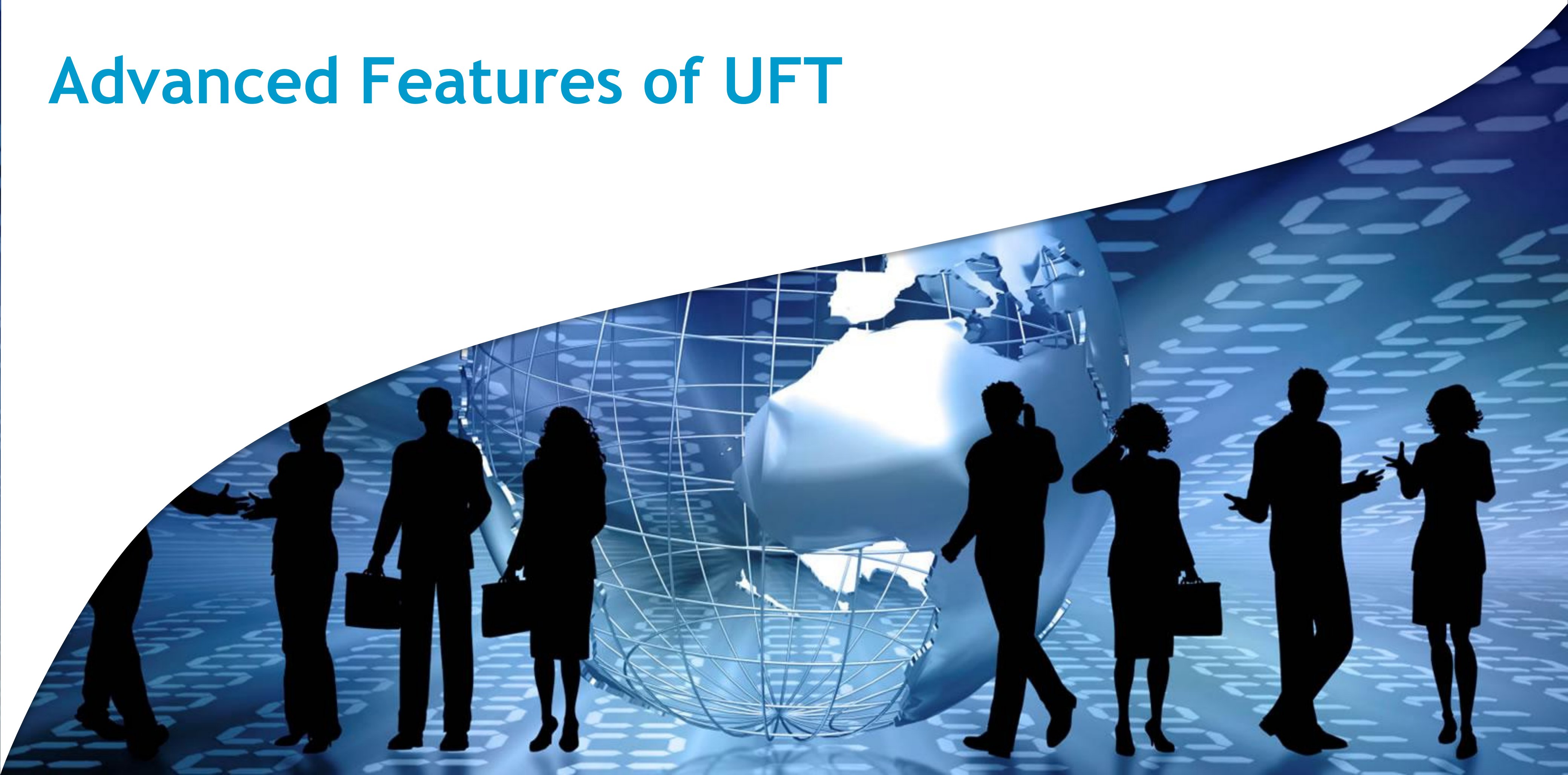
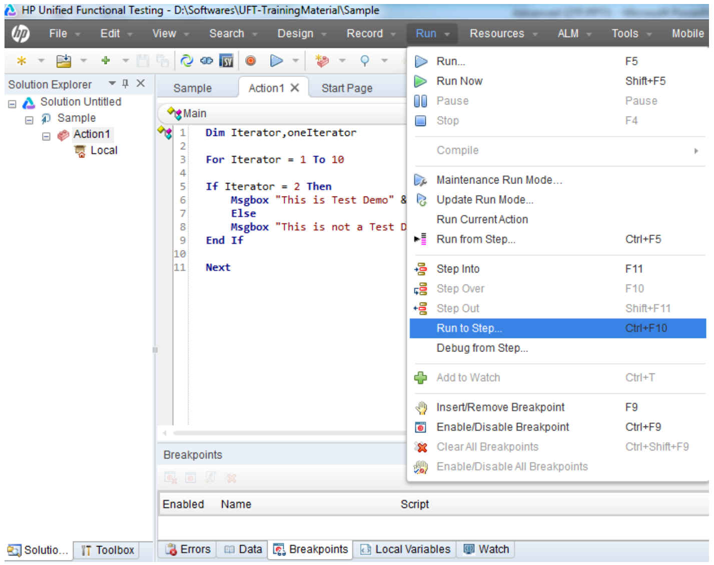
Advanced Features of UFT



Contents

Script Debugging

Script debugging



```
Dim Iterator,oneIterator

For Iterator = 1 To 10

If Iterator = 2 Then
    Msgbox "This is Test Demo" &
    Else
    Msgbox "This is not a Test D
End If

Next
```

• Script debugging is a process, involving careful examination of the code line by line while executing the script with an objective to see the actions performed by the script at every step.

• This is required to fix a script which does not perform as expected.

• Debugging is a process of eliminating the bugs in UFT scripts.

• For starting the debugging process, go to “Run" menu and select the desired debug process.

• We can start debugging from the existing position of the cursor or we can execute the script up to the cursor position in the debug mode

Script debugging

There are three types of debugging processes:

1) Debugging by "Step Into": When we select "Step Into" option, we can see if a function being executed is performing as expected.

This will open the function desired to be debugged in "Read Only" mode and we can keep on hitting the “F11” key on the keyboard to view the execution of every line of the function.

2) Debugging by "Step Over”: This option is selected when we are sure that the function is performing as expected & we don't want to view the execution of the function.

We can hit "F10" key to execute the entire function without stopping and will stop for our next command at the beginning of the next line after the function call.

3) Debugging by "Step Out”: This option is selected when we are in the function debug and we sure that the function is performing as expected & we don't want to debug the execution of the entire function.

We can hit Shift +"F10" keys to execute the remaining statements in the function without stopping and will stop for our next command at the beginning of the next line after the function call.

Debug Commands

- Run to Step :

  You can instruct QuickTest to run from the beginning of the test or action (Expert View only)—or from the current location in the test or action—and to stop at a particular step. This is similar to adding a temporary breakpoint to a step. For example, if you want to begin debugging your test or action from a particular step, you may want to run your test or action to that step, as this opens your application to the relevant location

- Debug from Step :

  You can instruct QuickTest to begin your debug session from a particular step instead of beginning the run at the start of the test or action. Before you start debugging from a specific step, make sure that the application is open to the location where you want to start debugging. You can begin debugging from a specific step in your test or action when editing a test or action

- Run from Step :

  You can use the Run from Step option to run a selected part of your component from the selected step to the end of the component. This enables you to check a specific section of your application or to confirm that a certain part of your component runs smoothly

Debug Viewer

The Debug Viewer pane includes the following tabs:

- **Watch Tab:**

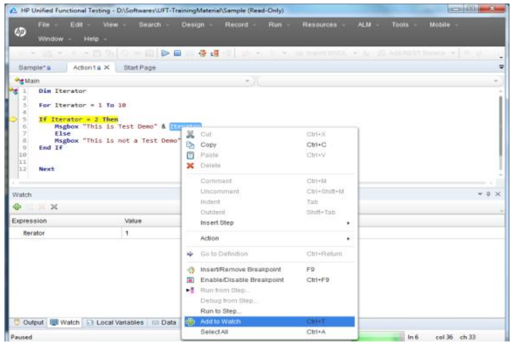
Displays the current values and types of variables and VBScript expressions that you add to the Watch tab, and enables you to modify the values of displayed variables and properties.

- **Local Variables Tab:**

This debug pane displays the current values and types of all variables in current the context of your document.

- **Breakpoints Tab:**

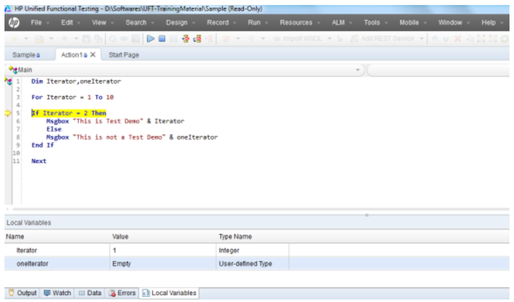
This debug pane enables you to view information about breakpoints inserted into your GUI actions, scripted GUI components, function libraries or user code files and navigate directly to the breakpoint location in the relevant document.


HP Unified Functional Testing - D:\Softwares\UFT-TrainingMaterial\Sample (Read-Only)

```
Dim Iterator

For Iterator = 1 To 10

If Iterator = 2 Then
    Msgbox "This is Test Demo" & Iterator
    Else
    Msgbox "This is not a Test Demo"
End If


Next
```

Watch

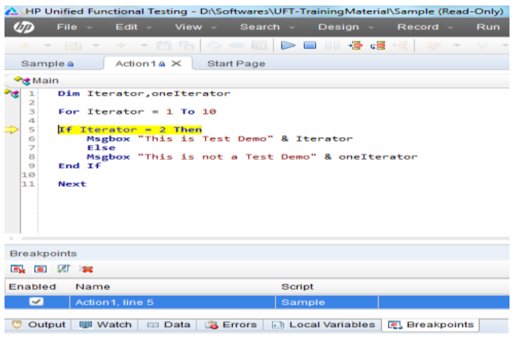
Expression	Value
Iterator	1


HP Unified Functional Testing - D:\Softwares\UFT-TrainingMaterial\Sample (Read-Only)

```
Dim Iterator,oneIterator

For Iterator = 1 To 10

If Iterator = 2 Then
    Msgbox "This is Test Demo" & Iterator
    Else
    Msgbox "This is not a Test Demo" & oneIterator
End If

Next
```

Local Variables

Name	Value	Type Name
Iterator	1	Integer
oneIterator	Empty	User-defined Type


HP Unified Functional Testing - D:\Softwares\UFT-TrainingMaterial\Sample (Read-Only)

```
Dim Iterator,oneIterator

For Iterator = 1 To 10

If Iterator = 2 Then
    Msgbox "This is Test Demo" & Iterator
    Else
    Msgbox "This is not a Test Demo" & oneIterator
End If

Next
```

Breakpoints

Enabled	Name	Script
✓	Action1, line 5	Sample

Debug Menu Commands

- Pause :
  Pauses the Run/debug session

- Add to Watch :
  Adds the selected item to the Watch tab.

- Insert/Remove Breakpoint :
  Sets or clears a breakpoint in the test

- Enable/Disable Breakpoint :
  Enables or disables a breakpoint in the test

- Clear All Breakpoints :
  Deletes all breakpoints in the test

- Enable/Disable All Breakpoints :
  Enables or disables all breakpoints in the test

Object Repository

How Quick Test Recognizes Objects

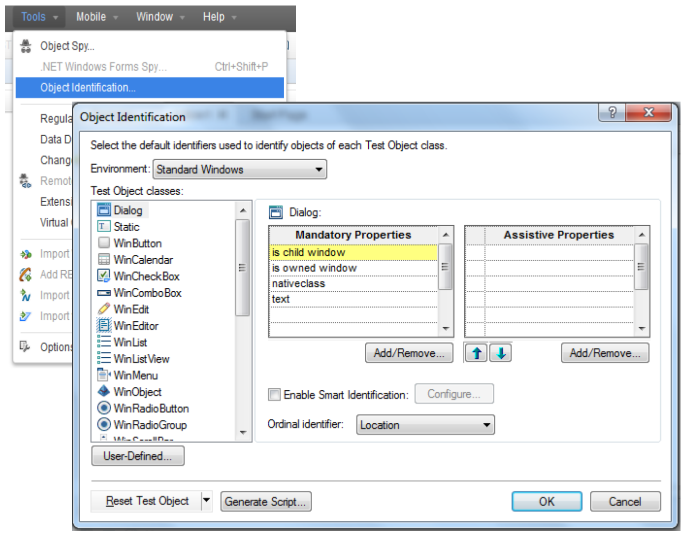
For each object class, UFT has a default set of properties that it always learns

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

Usually, only a few properties are needed to uniquely identify an object.

Visual Relation Identifier

Visual Relation Identifiers allow you to identify fields in your application based on other objects that are always near them



Object Identification

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

Environment: Standard Windows

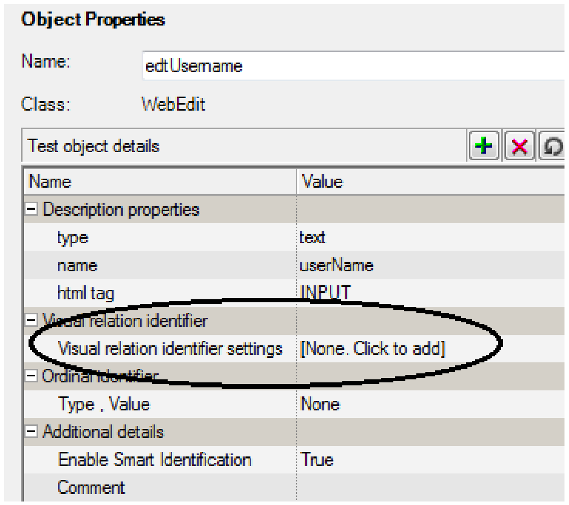
Test Object classes: Dialog, Static, WinButton, WinCalendar, WinCheckBox, WinComboBox, WinEdit, WinEditor, WinList, WinListView, WinMenu, WinObject, WinRadioButton, WinRadioGroup

Dialog:

Mandatory Properties	Assistive Properties
is child window	
is owned window	
nativeclass	
text	

Enable Smart Identification: [ ]

Ordinal identifier: Location



Object Properties

Name: edtUsername

Class: WebEdit

Test object details

Name	Value
Description properties	
type	text
name	userName
html tag	INPUT
Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	True
Comment	

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 QuickTest 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 QuickTest to create a unique description when the mandatory and assistive properties are not sufficient to do so

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

- 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

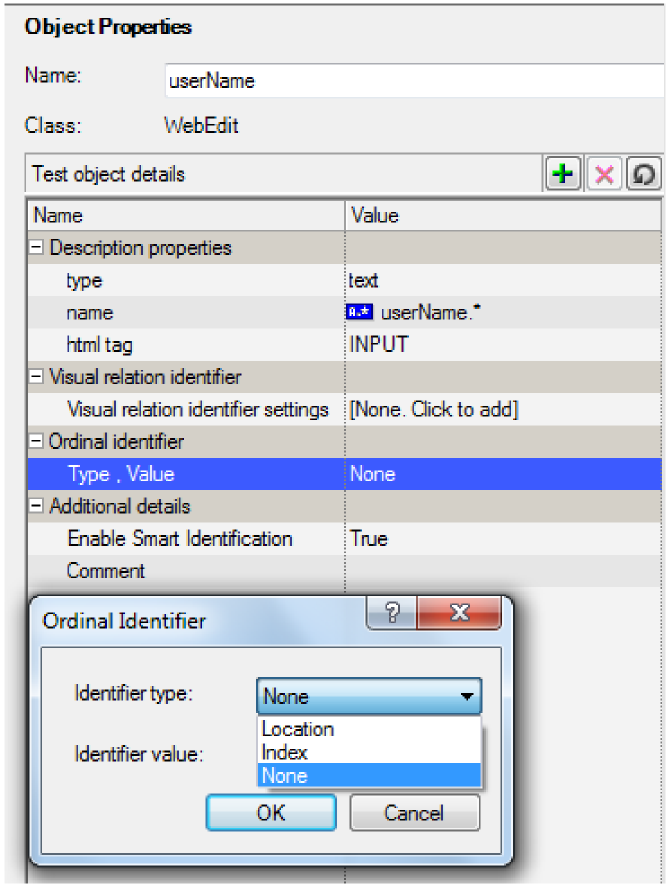
Ordinal Identifier

QuickTest 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.



Object Properties

Name: userName

Class: WebEdit

Test object details

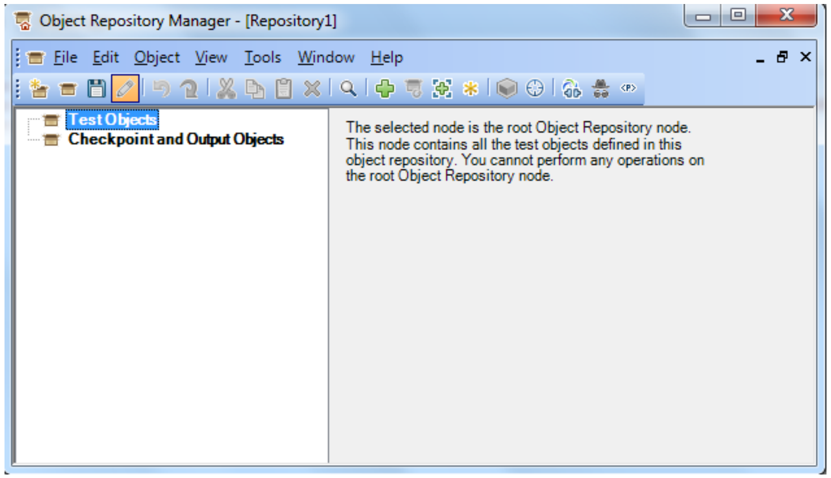
Name	Value
⊟ Description properties	
type	text
name	userName.*
html tag	INPUT
⊟ Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
⊟ Ordinal identifier	
Type , Value	None
⊟ Additional details	
Enable Smart Identification	True
Comment	

Ordinal Identifier

Identifier type: None

Identifier value:

Object Repository Manager


Object Repository Manager - [Repository1]
File Edit Object View Tools Window Help
Test Objects
Checkpoint and Output Objects

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.

• 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.

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

- Creating New Object Repositories
- Opening Object Repositories
- Saving Object Repositories
- Closing Object Repositories
- Managing Objects in Shared Object Repositories
- Managing Repository Parameters
- Modifying Object Details
- Locating Test Objects
- Performing Merge Operations
- Performing Import and Export Operations

Managing Repository Parameters

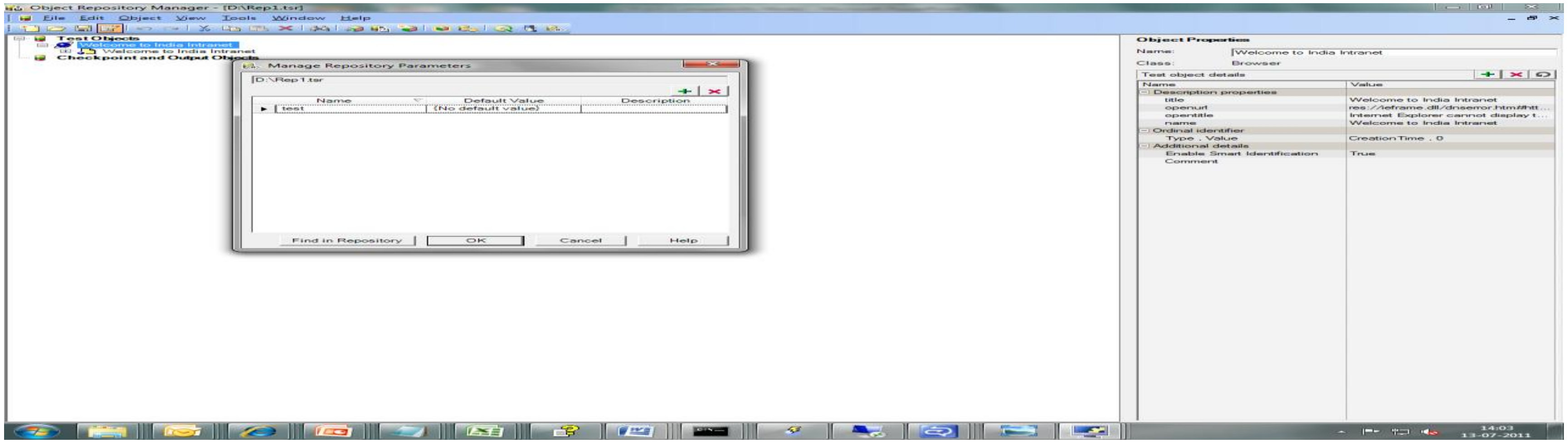
- Repository parameters enable you to specify the certain property values should be parameterized, but leave the actual parameterization to be defined in each test that is associated with the object repository that contains the parameterized identification property values

- 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

- 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



Object Repository Manager - [D:\Rep1.tsr]
File Edit Object View Tools Window Help
Test Objects
Welcome to India Intranet
Welcome to India Intranet
Checkpoint and Output Objects
Manage Repository Parameters
Find in Repository
OK
Cancel
Help
Object Properties
Test object details

D:\Rep1.tsr

Name	Default Value	Description
test	(No default value)	

Name: Welcome to India Intranet
Class: Browser

Name	Value
Description properties	
title	Welcome to India Intranet
openurl	res://ieframe.dll/dnserror.htm#htt...
opentitle	Internet Explorer cannot display t...
name	Welcome to India Intranet
Ordinal identifier	
Type , Value	CreationTime , 0
Additional details	
Enable Smart Identification	True
Comment	

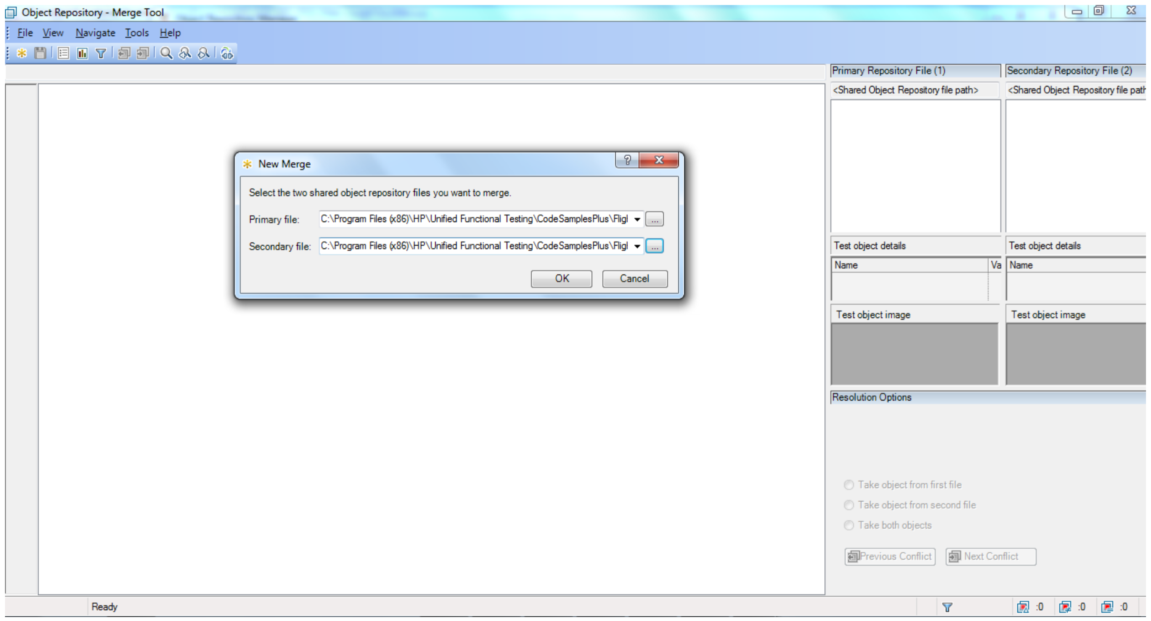
Object Repository Merge tool

- 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


Object Repository - Merge Tool
File
View
Navigate
Tools
Help
Primary Repository File (1)
Secondary Repository File (2)
<Shared Object Repository file path>
<Shared Object Repository file path
Test object details
Name
Va
Test object image
Resolution Options
Take object from first file
Take object from second file
Take both objects
Previous Conflict
Next Conflict
Ready
:0
:0
:0
New Merge
OK
Cancel

Select the two shared object repository files you want to merge.

Primary file: C:\Program Files (x86)\HP\Unified Functional Testing\CodeSamplesPlus\Fligl

Secondary file: C:\Program Files (x86)\HP\Unified Functional Testing\CodeSamplesPlus\Fligl

Object Repository Comparison Tool

- UFT 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

Performing Import and Export Operations

• You can import and export object repositories from and to XML files

• XML provides a structured, accessible format that enables you to make changes to object repositories using the XML editor of your choice and then import them back into QuickTest

• You can import and export files either from and to the file system or a Quality Center project (if QuickTest is connected to Quality Center).

Contents

Descriptive Programming

Descriptive Programming

What is Descriptive Programming?

- **Descriptive Programming (also known as Programmatic Description) provides a way to perform operations on objects that are not present in object repository.**
- **Programmatic description to instruct UFT to perform methods on objects without referring to the object repository.**
- **To do this, you provide QuickTest with a list of properties and values that QuickTest can use to identify the object or objects on which you want to perform a method.**

Descriptive programming – when and why?

Consider using DP in following cases:

• One of the very useful places where you can use Descriptive Programming is when the object properties in the Application Under Test (AUT) are dynamic in nature and need special handling to identify the object.

• Another place where DP can be of significant importance is when you are creating functions in an external file. You can use these function in various actions directly , eliminating the need of adding object(s) in object repository for each action. [If you use local object repository].

Example:

- Link Logout <User Name>

- Same objects on every page
  Example: Buttons – Next, Back, Cancel, OK

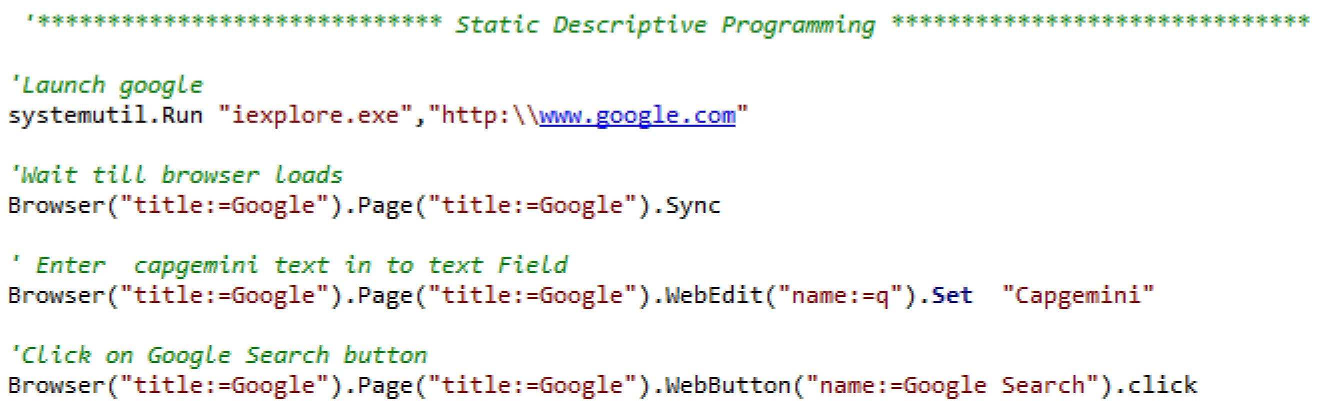
• Lots of similar objects on one page
  Example: table with many First & Last name text boxes

How to use DP?

There are two ways to use DP:

- Static Descriptive Programming
- Dynamic descriptive Programming

First Method...



```
'****************************** Static Descriptive Programming ******************************

'Launch google
systemutil.Run "iexplore.exe","http:\\www.google.com"

'Wait till browser loads
Browser("title:=Google").Page("title:=Google").Sync

' Enter  capgemini text in to text Field
Browser("title:=Google").Page("title:=Google").WebEdit("name:=q").Set  "Capgemini"

'Click on Google Search button
Browser("title:=Google").Page("title:=Google").WebButton("name:=Google Search").click
```

It’s the good example you know.

It’s the property:=value identification string

First Method...

That’s kinda restrictive

What if I want to use multiple identification properties?

Yes

No

First Method...

NO PROBLEM ☺

Browser("title:=Google", “name:=Google").Page("title:=Google").WebEdit("name:=q",” html tag:=INPUT”).**Set** "Capgemini"

You can use as many properties as you like

Second Method...

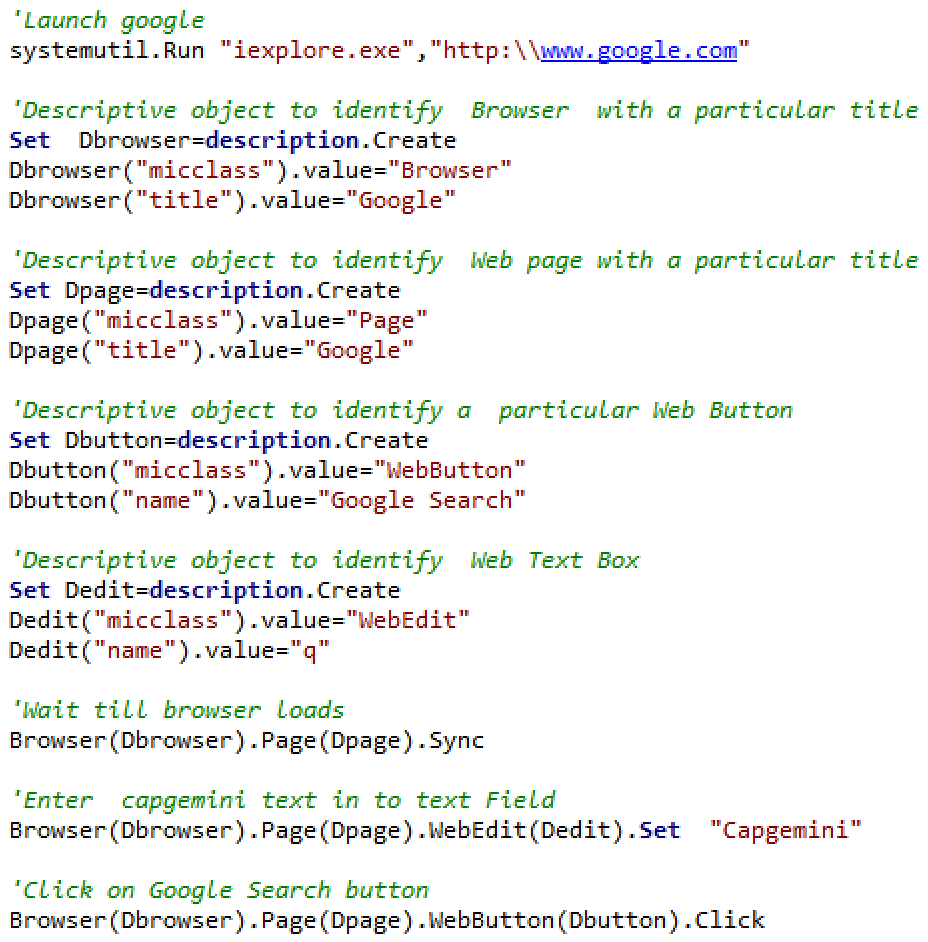
2

Throw the properties & values into a description object, and use it into the syntax.

Second Method...

Here also, all the values are interpreted as regular expressions. To turn it off, use

oDesc(“Property1”).RegularExpression = False



```
'Launch google
systemutil.Run "iexplore.exe","http:\\www.google.com"

'Descriptive object to identify  Browser  with a particular title
Set  Dbrowser=description.Create
Dbrowser("micclass").value="Browser"
Dbrowser("title").value="Google"

'Descriptive object to identify  Web page with a particular title
Set Dpage=description.Create
Dpage("micclass").value="Page"
Dpage("title").value="Google"

'Descriptive object to identify a  particular Web Button
Set Dbutton=description.Create
Dbutton("micclass").value="WebButton"
Dbutton("name").value="Google Search"

'Descriptive object to identify  Web Text Box
Set Dedit=description.Create
Dedit("micclass").value="WebEdit"
Dedit("name").value="q"

'Wait till browser loads
Browser(Dbrowser).Page(Dpage).Sync

'Enter  capgemini text in to text Field
Browser(Dbrowser).Page(Dpage).WebEdit(Dedit).Set  "Capgemini"

'Click on Google Search button
Browser(Dbrowser).Page(Dpage).WebButton(Dbutton).Click
```

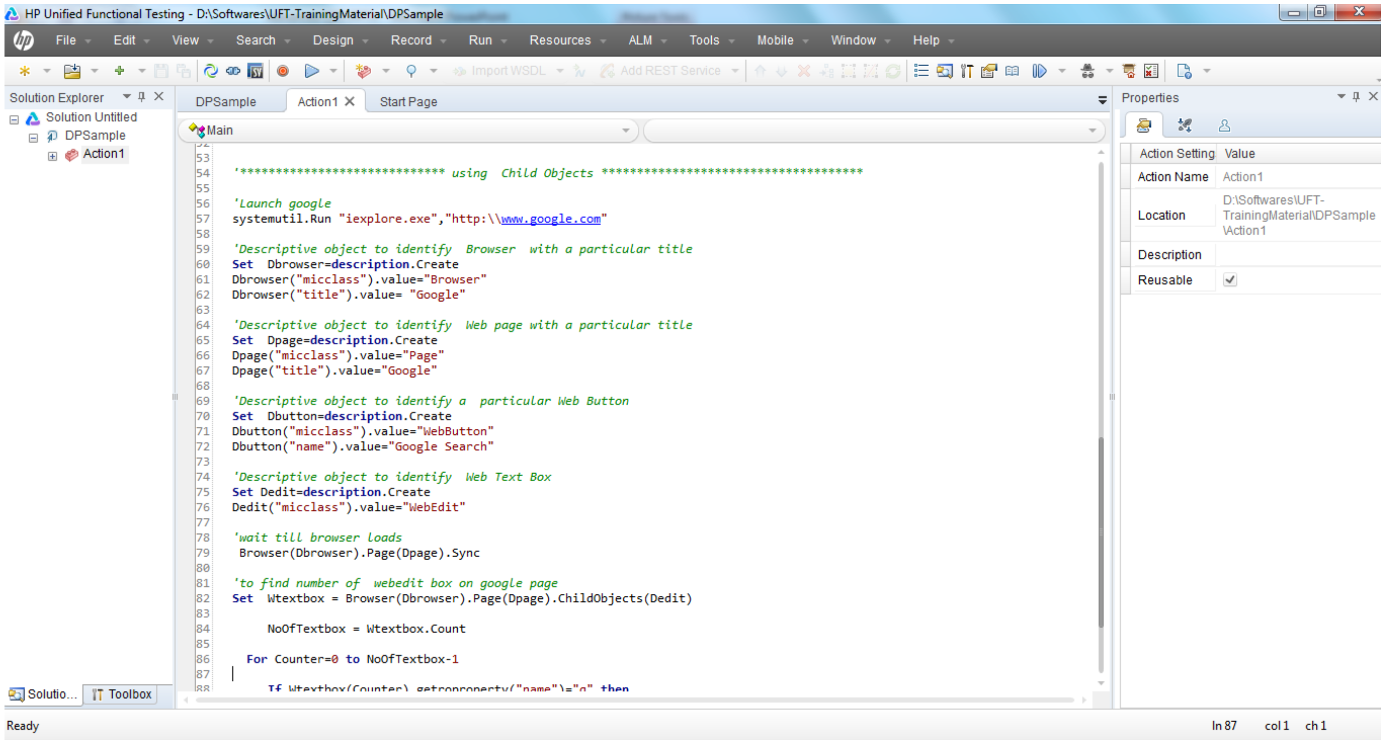
Descriptive Programming...

You can store the objects in collections.

When UFT finds two object which match the same description, it freezes ☹

Different ways to work with objects

Child Objects method – using Collection Object

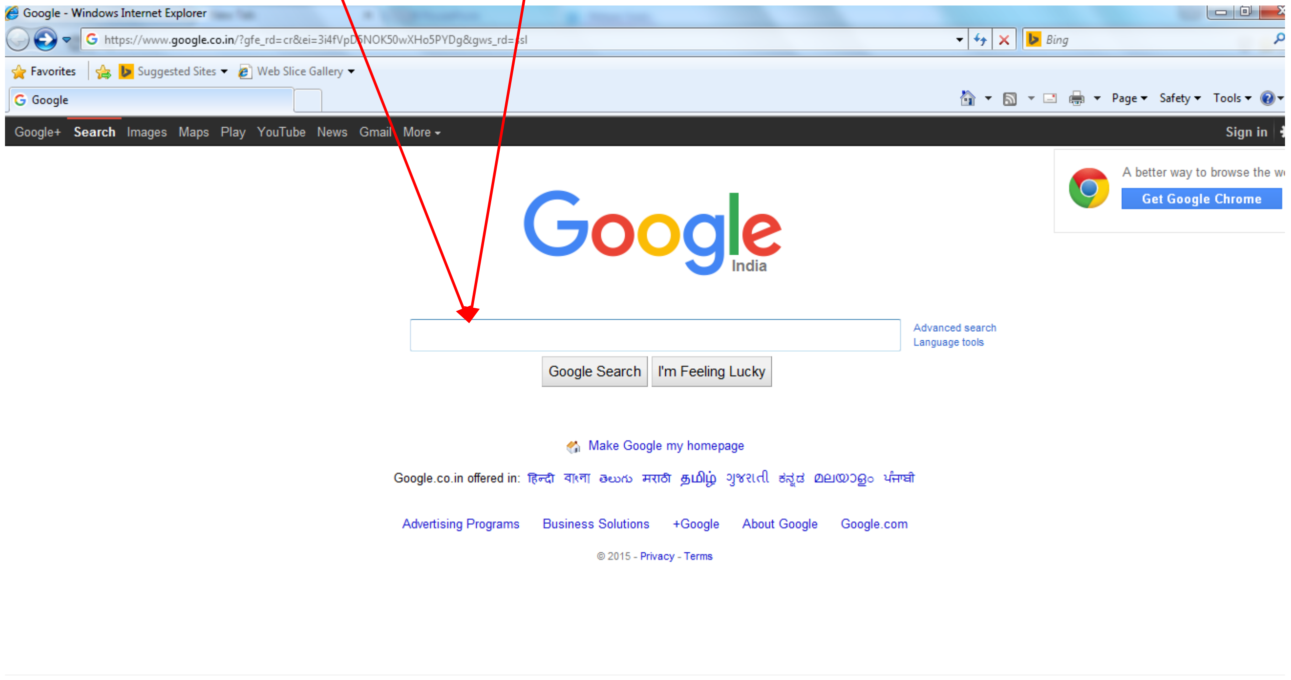


```
'***************************** using  Child Objects ************************************

'Launch google
systemutil.Run "iexplore.exe","http:\\www.google.com"

'Descriptive object to identify  Browser  with a particular title
Set  Dbrowser=description.Create
Dbrowser("micclass").value="Browser"
Dbrowser("title").value= "Google"

'Descriptive object to identify  Web page with a particular title
Set  Dpage=description.Create
Dpage("micclass").value="Page"
Dpage("title").value="Google"

'Descriptive object to identify a  particular Web Button
Set  Dbutton=description.Create
Dbutton("micclass").value="WebButton"
Dbutton("name").value="Google Search"

'Descriptive object to identify  Web Text Box
Set Dedit=description.Create
Dedit("micclass").value="WebEdit"

'wait till browser loads
 Browser(Dbrowser).Page(Dpage).Sync

'to find number of  webedit box on google page
Set  Wtextbox = Browser(Dbrowser).Page(Dpage).ChildObjects(Dedit)

     NoOfTextbox = Wtextbox.Count

  For Counter=0 to NoOfTextbox-1

     If Wtextbox(Counter).getroproperty("name")="q" then
```

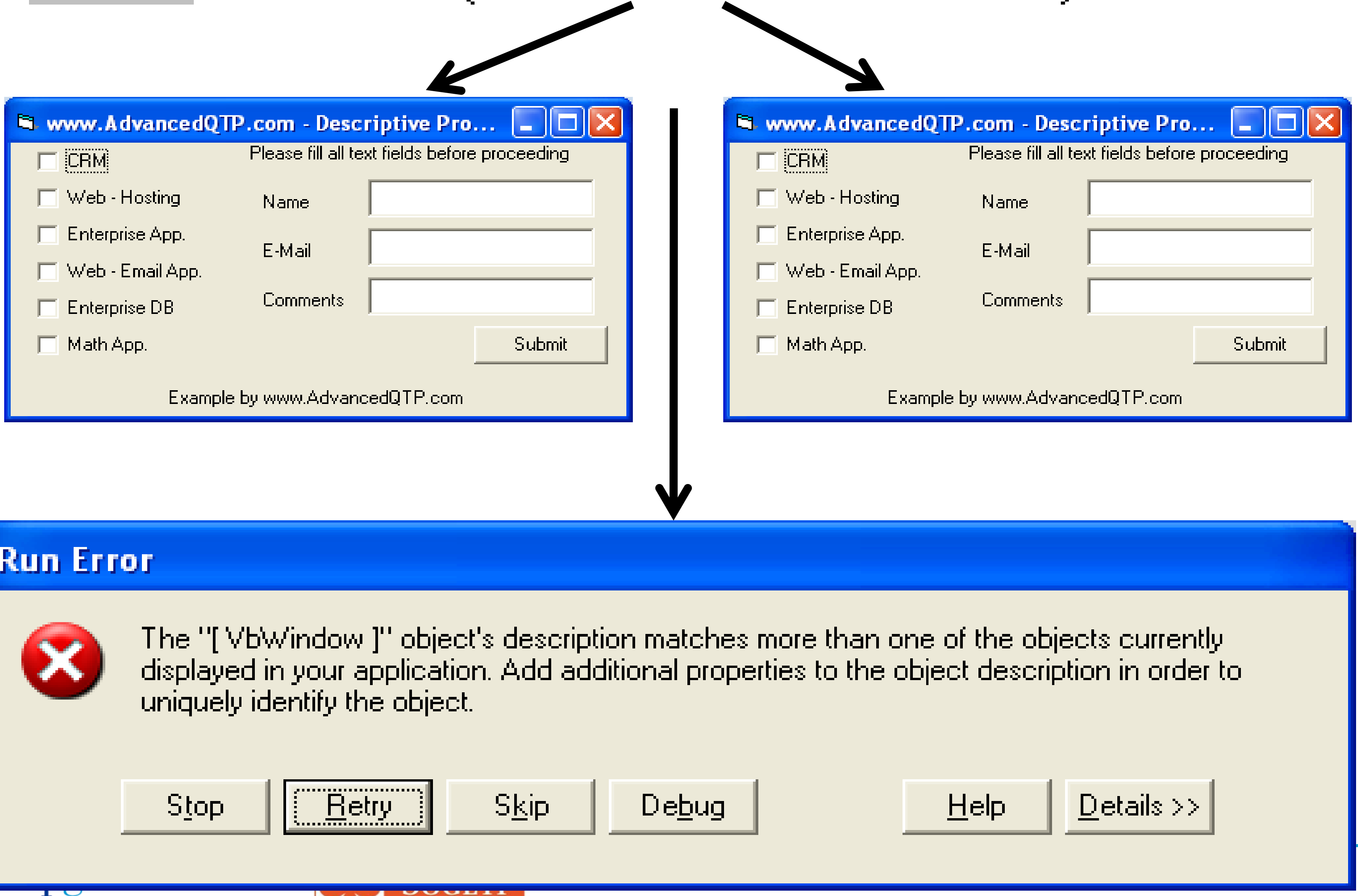
Action Setting	Value
Action Name	Action1
Location	D:\Softwares\UFT-TrainingMaterial\DPSample\Action1
Description	
Reusable	✓

ChildObjects method - using Collection Object

Search field is populated

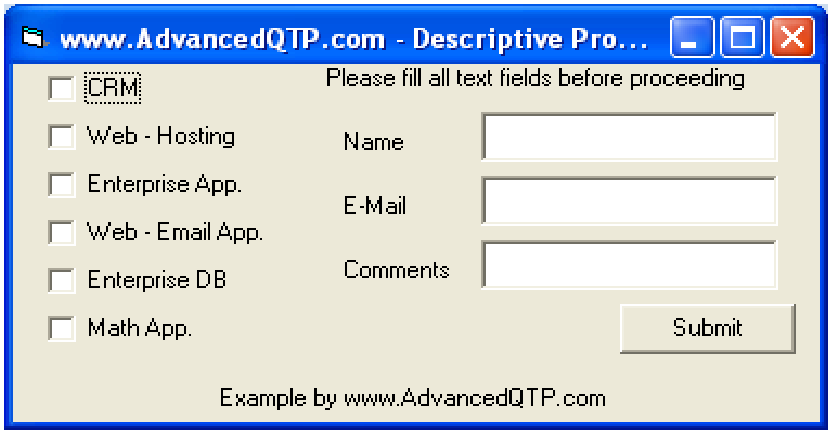
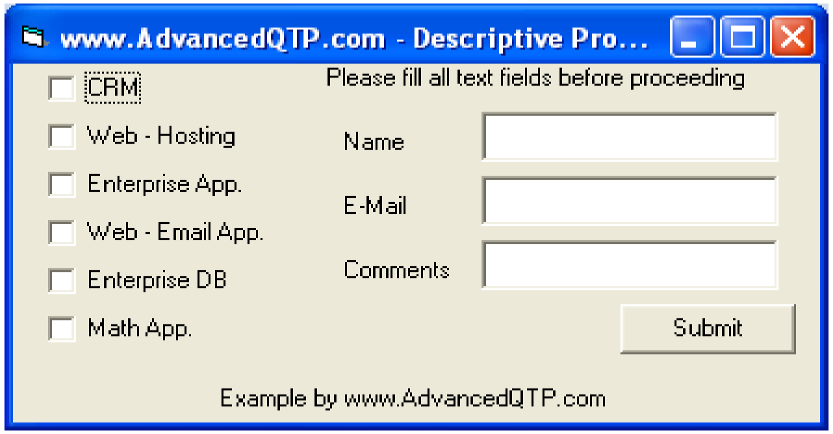


```
1:      VBWindow("title:=.*AdvancedQTP.*").Maximize
```


www.AdvancedQTP.com - Descriptive Pro...
CRM
Web - Hosting
Enterprise App.
Web - Email App.
Enterprise DB
Math App.
Please fill all text fields before proceeding
Name
E-Mail
Comments
Submit
Example by www.AdvancedQTP.com
www.AdvancedQTP.com - Descriptive Pro...
CRM
Web - Hosting
Enterprise App.
Web - Email App.
Enterprise DB
Math App.
Please fill all text fields before proceeding
Name
E-Mail
Comments
Submit
Example by www.AdvancedQTP.com
Run Error
Stop
Retry
Skip
Debug
Help
Details >>

```
The "[ VbWindow ]" object's description matches more than one of the objects currently
displayed in your application. Add additional properties to the object description in order to
uniquely identify the object.
```

```
VBWindow("title:=.*AdvancedQTP.*", "index:=0").Maximize
```

www.AdvancedQTP.com - Descriptive Pro...

CRM	Please fill all text fields before proceeding	
Web - Hosting	Name	
Enterprise App.	E-Mail	
Web - Email App.	Comments	
Enterprise DB		
Math App.		Submit

Example by www.AdvancedQTP.com

www.AdvancedQTP.com - Descriptive Pro...

CRM	Please fill all text fields before proceeding	
Web - Hosting	Name	
Enterprise App.	E-Mail	
Web - Email App.	Comments	
Enterprise DB		
Math App.		Submit

Example by www.AdvancedQTP.com

DP has a magic property: “index”, which allows us to tell the double objects apart

Index is a zero-based counter

TO, RO and .Object

- .GetTOproperty/SetToProperty refers to the properties stored in OR
- .GetROProperty property refers to the AUT Object property (Run-time)
- .Object.<property/method> refers to the AUT Object NATIVE properties/methods

Object Repository vs. Descriptive Programming –what to use?

- There really is no "best way"

- Use the method that gives your company the best ROI, whether that be Object Repository (OR), Descriptive Programming (DP) or a mixture of both

OR Pros and Cons

PROS:

- GUI Front end to examine all the objects in the repository
- Highlight in Application feature is great tool to walk the object tree
- No need to modify the script when object properties changes
- Easy to identify objects in AUT by Object Logical names
- Can be created independently from scripts

OR Pros and Cons

CONS:

- Additional layer to maintain
- Unnecessary objects can be created
- Multiple users cannot concurrently save/write to the shared OR
- It won’t eliminate the need for Descriptive Programming in most of cases

DP Pros and Cons

PROS:

- It’s a white box
- Compatible with different UFT versions
- Code portability is high
- Easy to mass update

DP Pros and Cons

CONS:

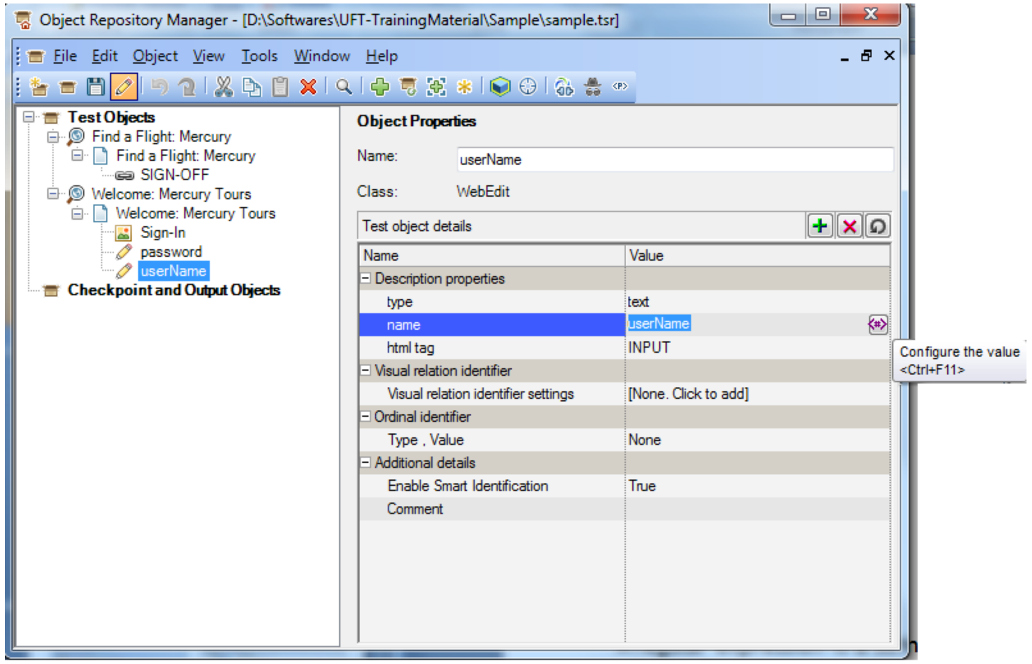
- Lower Code Readability and requires more comments, like “what object is accessed”
- Potentially slower to create
- To highlight an object in the application requires utilizing the “Highlight” method

Regular Expressions

Regular Expressions

- Regular Expressions can be used to identify the objects in the application with varying values or names or titles.
- Regular expressions can be added by,
    - Defining the property values of an object in
      dialog boxes or in programmatic descriptions
    - Parameterize a step
    - Creating checkpoints with varying values
- A Regular Expression is a pattern of text that consists of
    - Alphabets - letters a through z
    - Special characters known as Metacharacters.
    - Numbers
- The pattern describes one or more strings to match when searching a body of text.
- The Regular Expression serves as a template for matching a character pattern to the string being searched.

Regular Expressions



Object Repository Manager - [D:\Softwares\UFT-TrainingMaterial\Sample\sample.tsr]
File Edit Object View Tools Window Help
Test Objects
Find a Flight: Mercury
Find a Flight: Mercury
SIGN-OFF
Welcome: Mercury Tours
Welcome: Mercury Tours
Sign-In
password
userName
Checkpoint and Output Objects
Configure the value <Ctrl+F11>

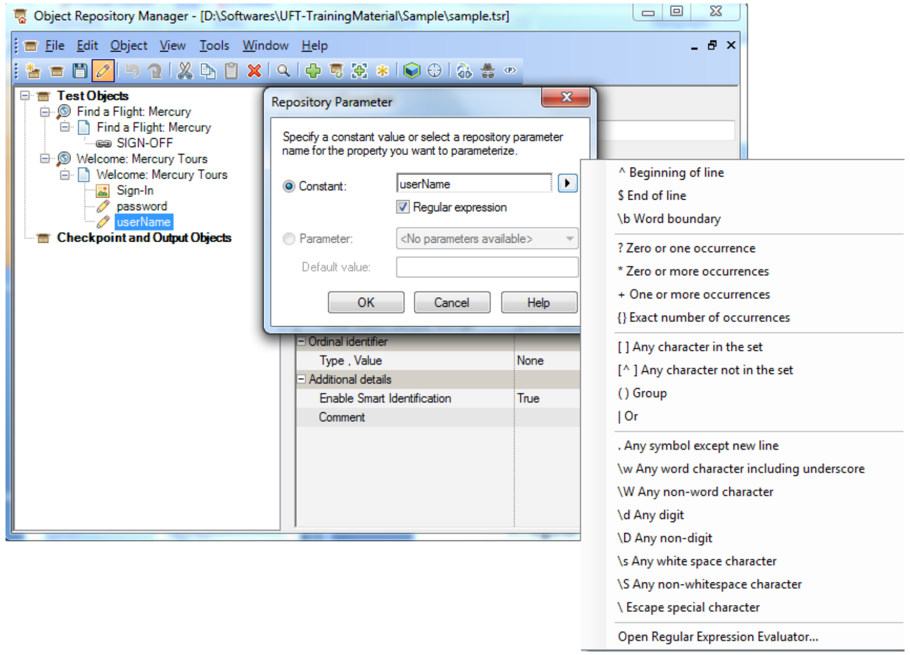
Object Properties

Name: userName

Class: WebEdit

Test object details

Name	Value
Description properties	
type	text
name	userName
html tag	INPUT
Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	True
Comment	



Object Repository Manager - [D:\Softwares\UFT-TrainingMaterial\Sample\sample.tsr]
File Edit Object View Tools Window Help
Test Objects
Find a Flight: Mercury
Find a Flight: Mercury
SIGN-OFF
Welcome: Mercury Tours
Welcome: Mercury Tours
Sign-In
password
userName
Checkpoint and Output Objects
Repository Parameter
OK
Cancel
Help
Open Regular Expression Evaluator...

Specify a constant value or select a repository parameter name for the property you want to parameterize.

[✓ Constant:] userName

[✓] Regular expression

Parameter: <No parameters available>

Default value:

Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	True
Comment	

^	Beginning of line
$	End of line
\b	Word boundary
?	Zero or one occurrence
*	Zero or more occurrences
+	One or more occurrences
{}	Exact number of occurrences
[ ]	Any character in the set
[^]	Any character not in the set
( )	Group
|	Or
.	Any symbol except new line
\w	Any word character including underscore
\W	Any non-word character
\d	Any digit
\D	Any non-digit
\s	Any white space character
\S	Any non-whitespace character
\	Escape special character

A regular expression is a string that describes or matches a set of strings. It is often called a pattern as it describes set of strings.

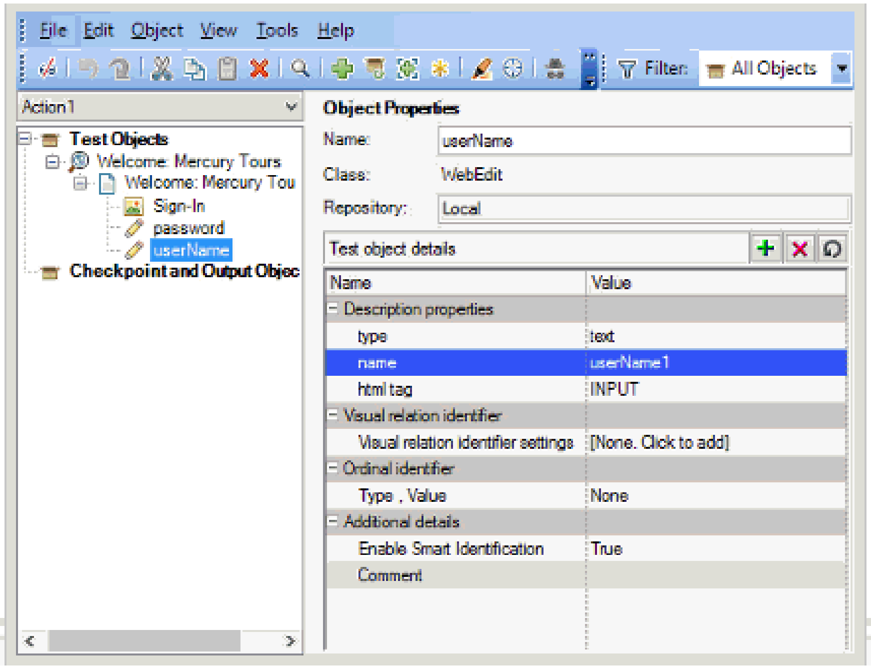
OR

A regular expression is a special text string for describing a search pattern.

Use Of Regular Expressions

It is used to identify "objects“ & "text strings" with varying values

- Use Regular Expressions only for values of type string.



Object Properties

Name: userName

Class: WebEdit

Repository: Local

Test object details

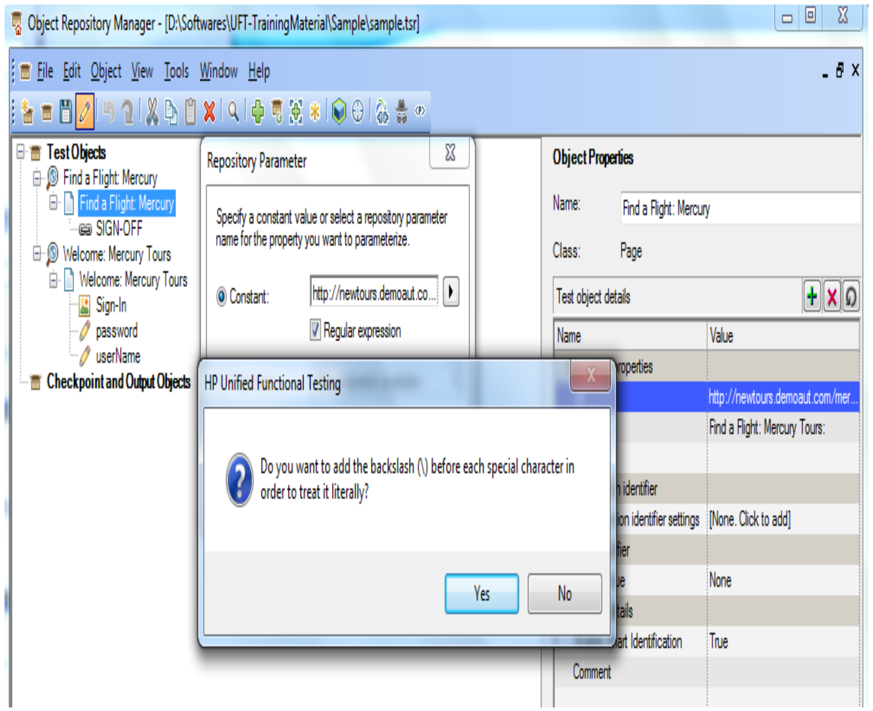
Name	Value
Description properties	
type	text
name	userName1
html tag	INPUT
Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	True
Comment	

- Use Regular Expressions for Property Values

```
 Browser("Welcome:MercuryTours").Page("Welcome:MercuryTours").WebEdit("type:=text","name:=userName.*","html
tag:=INPUT").Set "mercury"
```

Regular Expressions - Object Repository

Regular Expressions - for URL



Object Repository Manager - [D:\Softwares\UFT-TrainingMaterial\Sample\sample.tsr]

File Edit Object View Tools Window Help

- Test Objects
  - Find a Flight: Mercury
    - Find a Flight: Mercury
      - SIGN-OFF
  - Welcome: Mercury Tours
    - Welcome: Mercury Tours
      - Sign-In
      - password
      - userName
- Checkpoint and Output Objects

Repository Parameter

Specify a constant value or select a repository parameter name for the property you want to parameterize.

Constant: http://newtours.demoaut.co...

☑ Regular expression

HP Unified Functional Testing

Do you want to add the backslash (\) before each special character in order to treat it literally?

Yes No

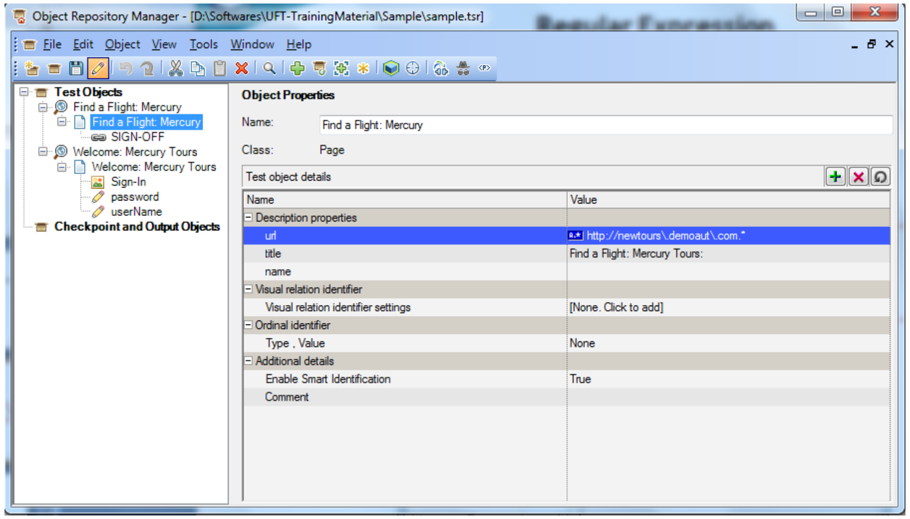
Object Properties

Name: Find a Flight: Mercury

Class: Page

Test object details

Name	Value
...roperties	
	http://newtours.demoaut.com/mer...
	Find a Flight: Mercury Tours:
...identifier	
...ion identifier settings	[None. Click to add]
...fier	
...e	None
...tails	
...art Identification	True
Comment	



Object Repository Manager - [D:\Softwares\UFT-TrainingMaterial\Sample\sample.tsr]

File Edit Object View Tools Window Help

- Test Objects
  - Find a Flight: Mercury
    - Find a Flight: Mercury
      - SIGN-OFF
  - Welcome: Mercury Tours
    - Welcome: Mercury Tours
      - Sign-In
      - password
      - userName
- Checkpoint and Output Objects

Object Properties

Name: Find a Flight: Mercury

Class: Page

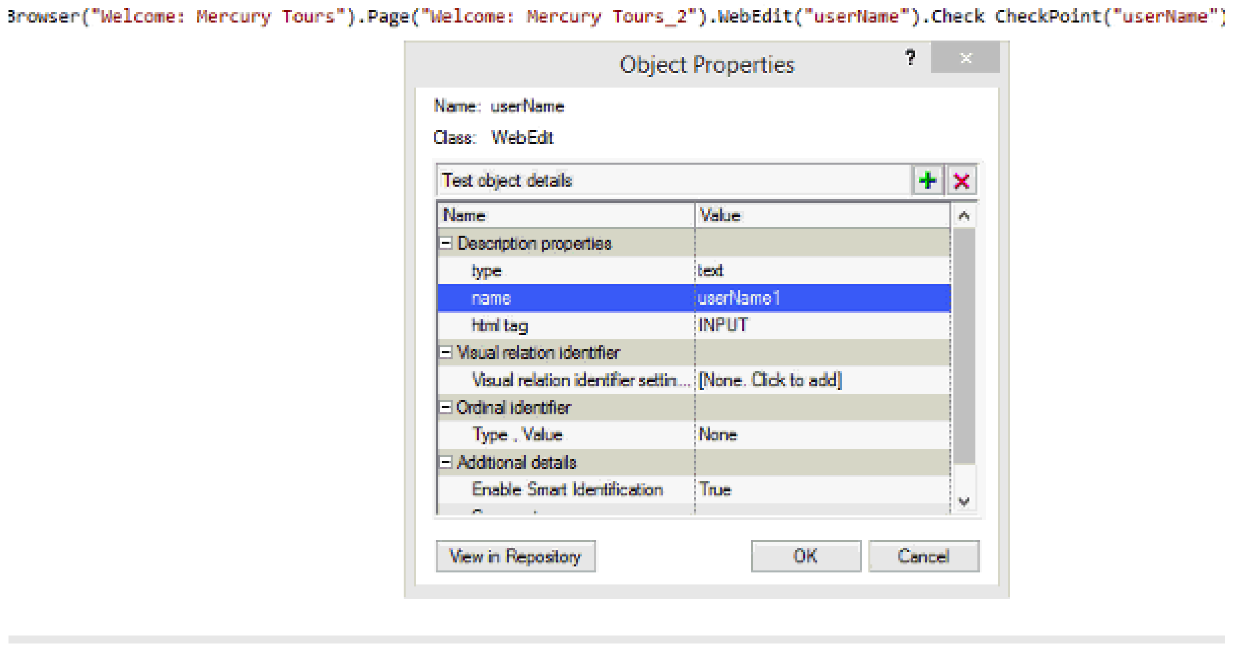
Test object details

Name	Value
Description properties	
url	http://newtours\.demoaut\.com.*
title	Find a Flight: Mercury Tours:
name	
Visual relation identifier	
Visual relation identifier settings	[None. Click to add]
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	True
Comment	

The Object Repository should look like this after making the necessary changes.

Use Of Regular Expressions

- Using Regular Expressions for Checkpoints



```
3rowser("Welcome: Mercury Tours").Page("Welcome: Mercury Tours_2").WebEdit("userName").Check CheckPoint("userName")
```

Object Properties

Name: userName

Class: WebEdit

Test object details

Name	Value
⊟ Description properties	
type	text
name	userName1
Html tag	INPUT
⊟ Visual relation identifier	
Visual relation identifier settin...	[None. Click to add]
⊟ Ordinal identifier	
Type , Value	None
⊟ Additional details	
Enable Smart Identification	True

View in Repository | OK | Cancel

For example,

In the example statement shown in the above, once you have inserted a checkpoint, right click on Checkpoint(“username”) and go to Checkpoint properties. Highlight the property value which you wish to make a regular expression.

Regular Expressions Characters

Using the Backslash Character (\ )

Matching Any Single Character (.)

Matching Any Single Character in a List ([xy])

Matching Any Single Character Not in a List ([^xy])

Matching Any Single Character within a Range ([x-y])

Matching Zero or More Specific Characters (*)

Matching One or More Specific Characters (+)

Matching Zero or One Specific Character (?)

Grouping Regular Expressions (( ))

Matching One of Several Regular Expressions (|)

Matching the Beginning of a Line (^)

Matching the End of a Line ($)

Matching Any Alphanumeric Character Including the Underscore(\w)

Matching Any Non-Alphanumeric Character (\W)

Regular Expressions – Object Repository

To handle windows with varying titles

The Fax Order screen in sample Fight Application is an example for a window with varying title.

Insert an order and playback the script.

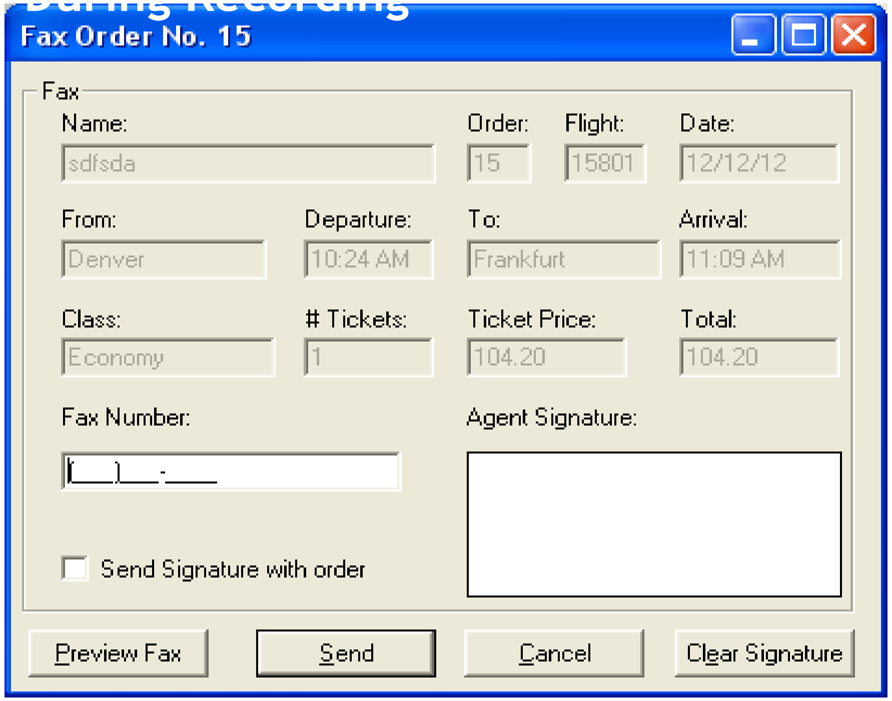
Follow the steps given below:

- Start Recording
- Insert an Order
- Open the Fax order - File - > Fax Order
- Close the fax order window
- Close the AUT
- Stop the recording
- Run the test

Regular Expressions

The fax order window name changes whenever a new order is inserted.

During Recording



Fax Order No. 15

Fax

Name:	Order:	Flight:	Date:
sdfsda	15	15801	12/12/12

From:	Departure:	To:	Arrival:
Denver	10:24 AM	Frankfurt	11:09 AM

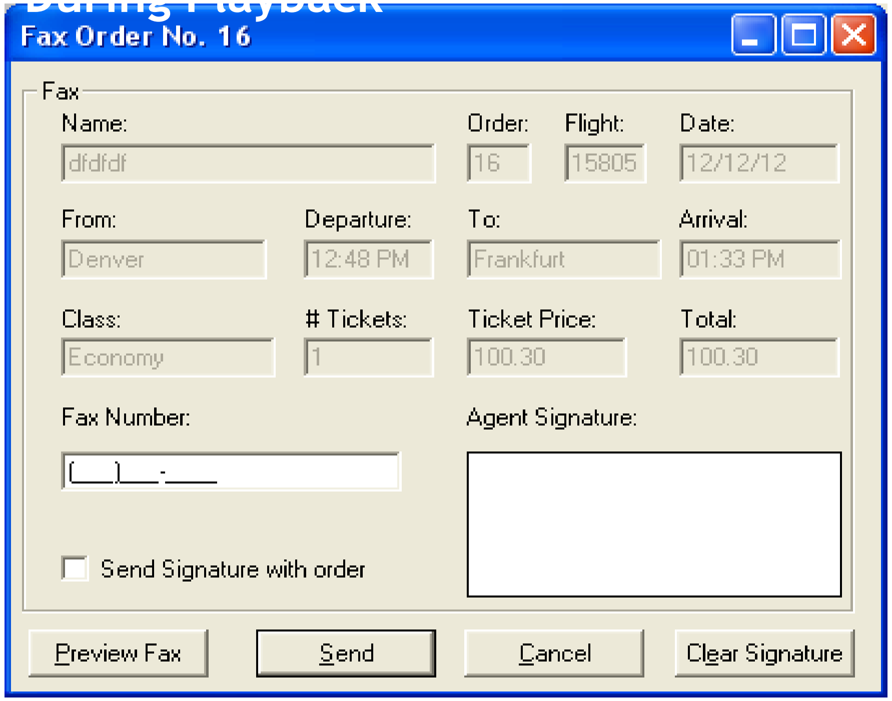
Class:	# Tickets:	Ticket Price:	Total:
Economy	1	104.20	104.20

Fax Number: (___)___-____

Agent Signature:

☐ Send Signature with order

During Playback



Fax Order No. 16

Fax

Name:	Order:	Flight:	Date:
dfdfdf	16	15805	12/12/12

From:	Departure:	To:	Arrival:
Denver	12:48 PM	Frankfurt	01:33 PM

Class:	# Tickets:	Ticket Price:	Total:
Economy	1	100.30	100.30

Fax Number: (___)___-____

Agent Signature:

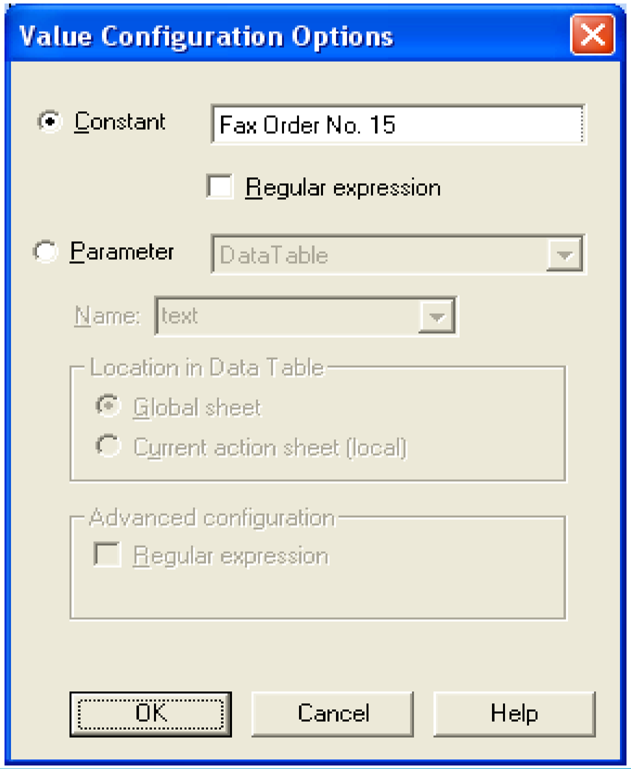
☐ Send Signature with order

Regular Expression - Object Repository

Select Description Properties - text

Click the Configure value button next to text

The following window will be displayed


Value Configuration Options

(● Constant) Fax Order No. 15

☐ Regular expression

○ Parameter DataTable

Name: text

Location in Data Table

● Global sheet

○ Current action sheet (local)

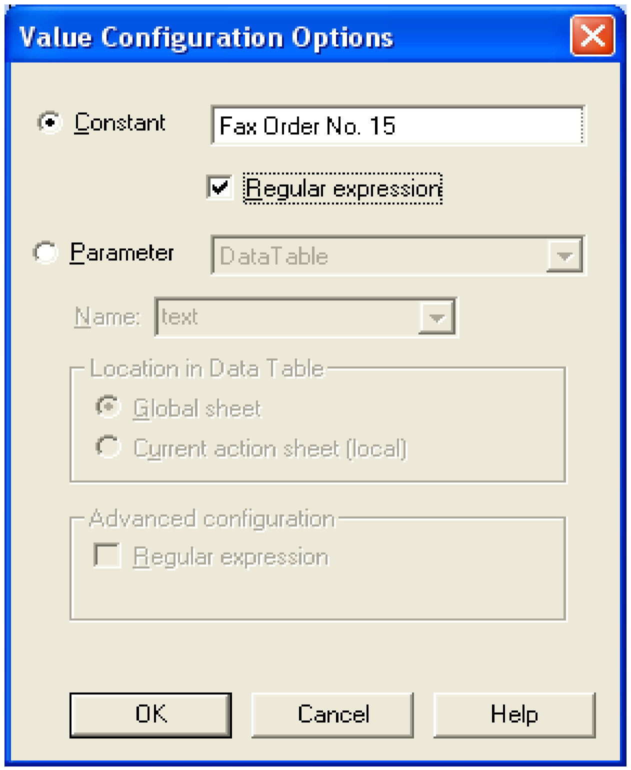
Advanced configuration

☐ Regular expression

Regular Expression

Change the name of the window as Fax Order No. .*

Click the Regular Expression Check box



Value Configuration Options

Constant [✓] Fax Order No. 15

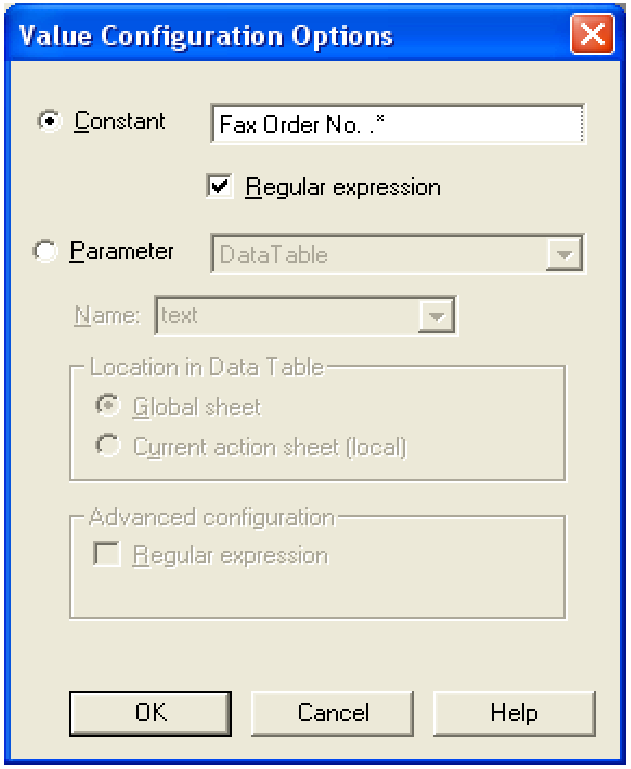
[✓] Regular expression

Parameter: DataTable

Name: text

Location in Data Table: [✓ Global sheet] Current action sheet (local)

Advanced configuration: [ ] Regular expression



Value Configuration Options

Constant [✓] Fax Order No. .*

[✓] Regular expression

Parameter: DataTable

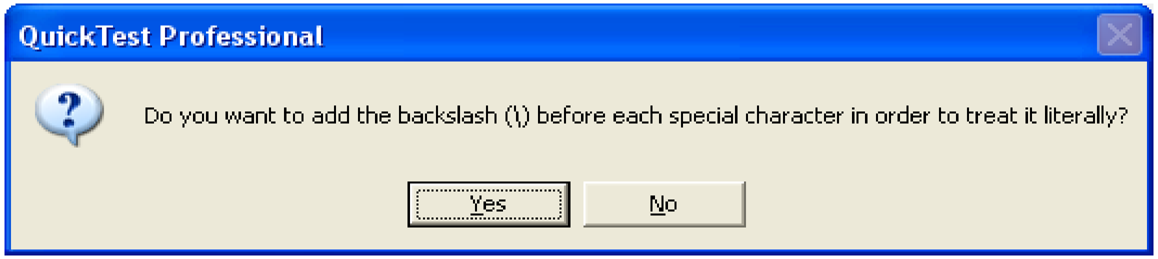
Name: text

Location in Data Table: [✓ Global sheet] Current action sheet (local)

Advanced configuration: [ ] Regular expression

Regular Expression

Click No in the msgbox displayed after closing the value configuration screen.


QuickTest Professional

Do you want to add the backslash (\) before each special character in order to treat it literally?

Virtual Objects

Virtual Objects

- Virtual Objects are objects that behaves like normal objects, but are not recognized by QuickTest.

- We can define these objects as Virtual Objects and map them to standard classes, such as a button or a check box.

- A Virtual Object collection is a group of virtual objects that is stored in the Virtual Object Manager under a descriptive name.

Defining a Virtual Object

- We define a Virtual Object using the Virtual Object Wizard.

- Using the Virtual Object Wizard, we can map a virtual object to a standard object class, specify the boundaries and the parent of the virtual object, and assign it a name.

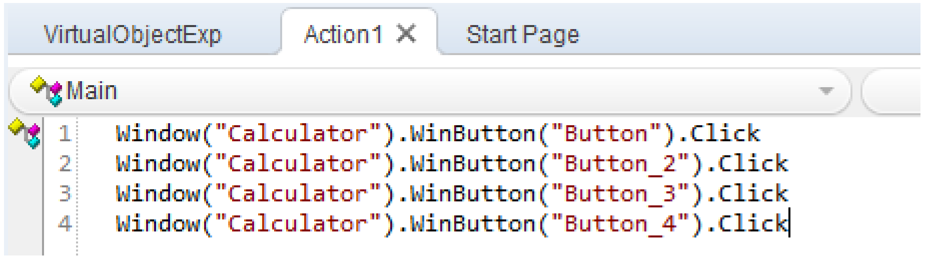
- Only those objects can be defined as Virtual Objects on which we can click or double-click and that record a Click or DblClick step. Otherwise, the virtual object is ignored.

Steps for Creating Virtual Object

- In UFT, choose Tools > Virtual Objects > New Virtual Object.
- Select a standard class to which you want to map your virtual object.
- Click Mark Object button. Use the crosshairs pointer to mark the area of the virtual object.
- An object in the object tree is assigned as the parent of the virtual object.
- Specify a name and a collection for the virtual object.

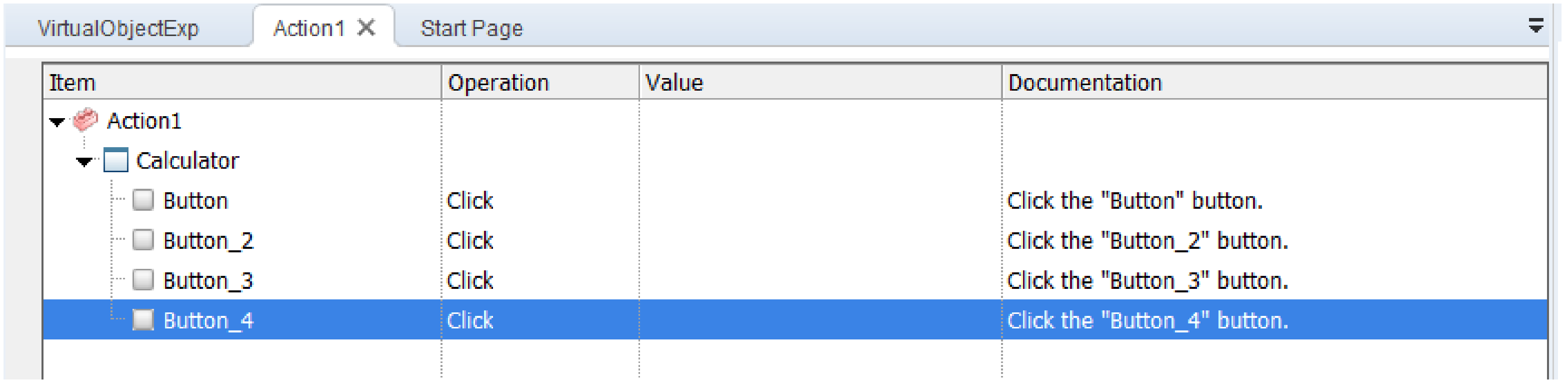
Case Study of Virtual Object

Let us consider the calculator application. Suppose we are recording the calculations that are performed in the calculator. The script will be recorded as follows.



```
Window("Calculator").WinButton("Button").Click
Window("Calculator").WinButton("Button_2").Click
Window("Calculator").WinButton("Button_3").Click
Window("Calculator").WinButton("Button_4").Click
```

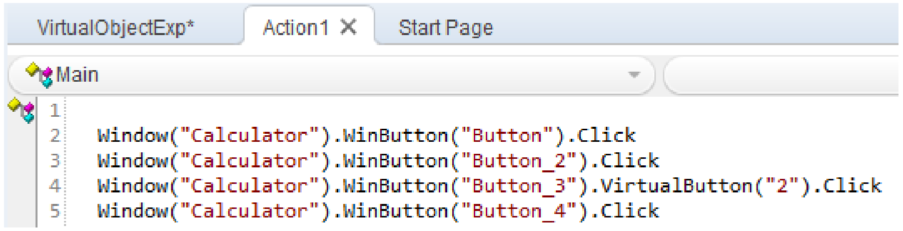
The corresponding Keyword View will be as shown below.



Item	Operation	Value	Documentation
Action1			
Calculator			
Button	Click		Click the "Button" button.
Button_2	Click		Click the "Button_2" button.
Button_3	Click		Click the "Button_3" button.
Button_4	Click		Click the "Button_4" button.

Case Study of Virtual Object (contd.)

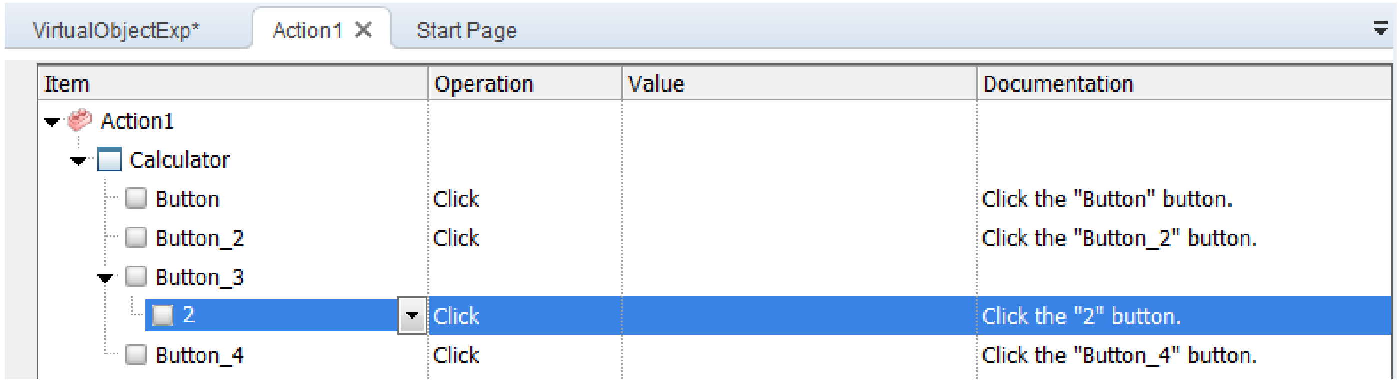
Suppose button ‘Button_3**’ is made a virtual object and we give the name ‘**2**’.Now, the script will be displayed as shown below. The button “**Button_3**” has been assigned as an virtual object.**


VirtualObjectExp* | Action1 | Start Page

Main

```
Window("Calculator").WinButton("Button").Click
Window("Calculator").WinButton("Button_2").Click
Window("Calculator").WinButton("Button_3").VirtualButton("2").Click
Window("Calculator").WinButton("Button_4").Click
```

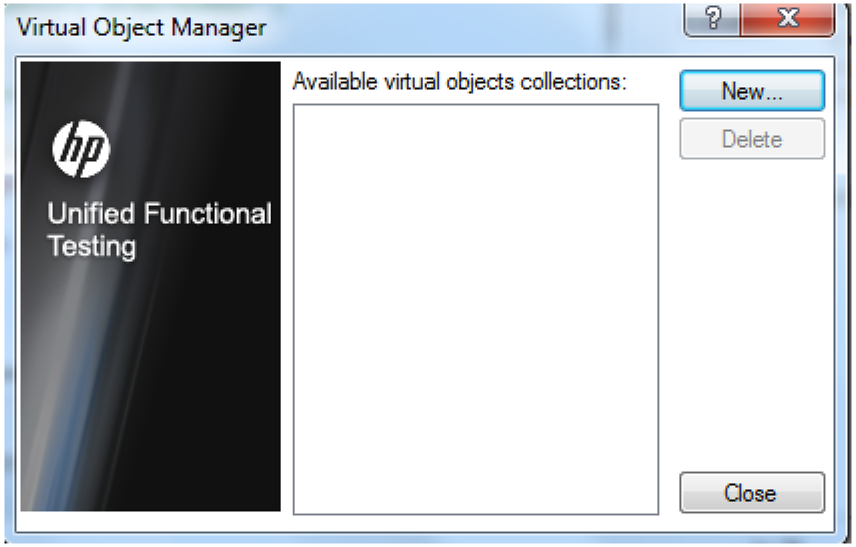
The corresponding Keyword View will be as follows. The virtual object can be identified by the symbol ‘v’ attached to the object.


VirtualObjectExp* | Action1 | Start Page

Item	Operation	Value	Documentation
Action1			
Calculator			
Button	Click		Click the "Button" button.
Button_2	Click		Click the "Button_2" button.
Button_3			
2	Click		Click the "2" button.
Button_4	Click		Click the "Button_4" button.

Removing Virtual Objects

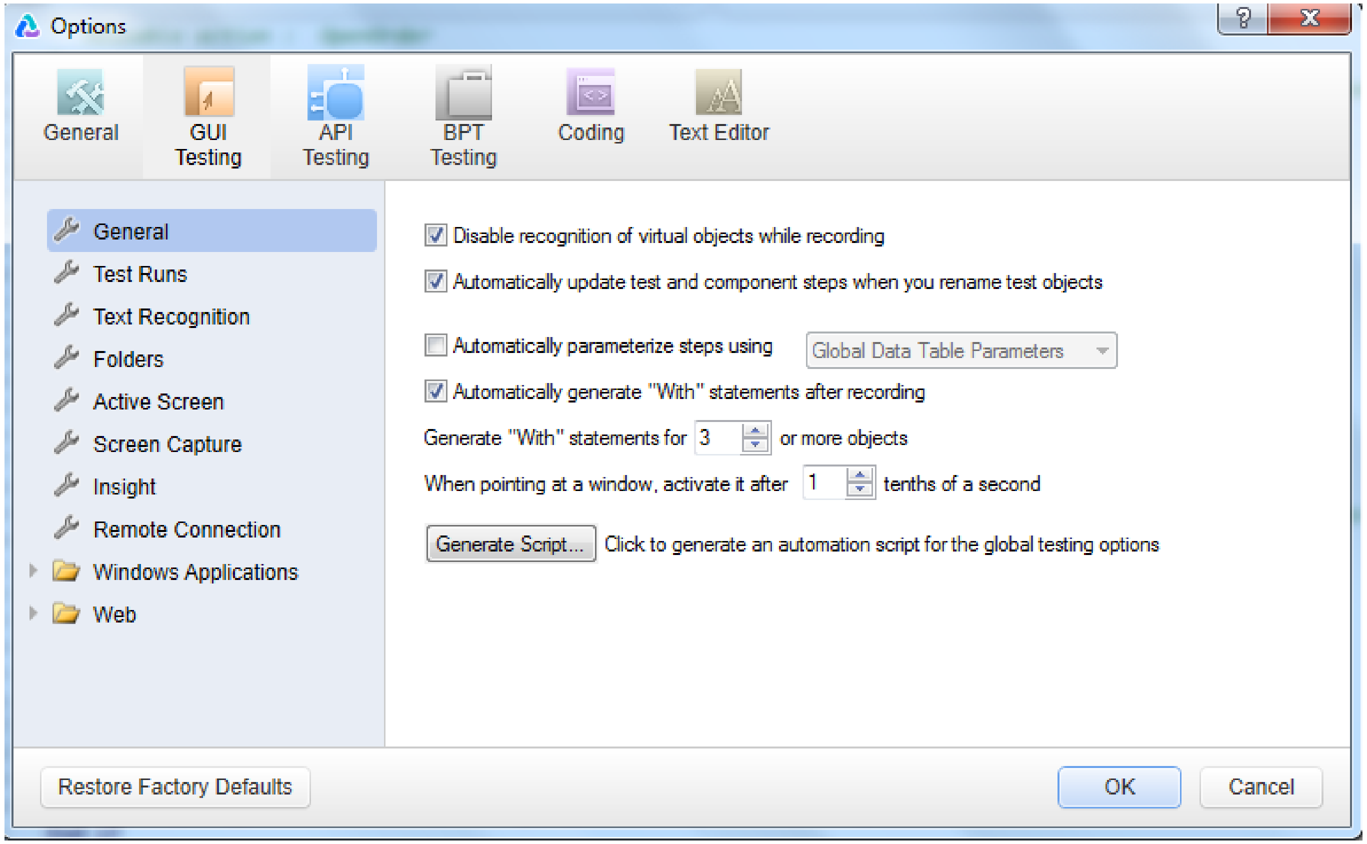
We can remove virtual objects from were test or component by deleting them from Virtual Object Manager that can be accessed from Tools→Virtual Object→ Virtual Object Manager


Virtual Object Manager
Unified Functional Testing
Available virtual objects collections:
New...
Delete
Close

Disabling Virtual Objects

Choose Tools > Options or click the Options toolbar button. The Options dialog box opens.

In the General tab, select the Disable recognition of Virtual Objects while recording



- [✓] Disable recognition of virtual objects while recording
- [✓] Automatically update test and component steps when you rename test objects
- [ ] Automatically parameterize steps using: Global Data Table Parameters
- [✓] Automatically generate "With" statements after recording

Generate "With" statements for 3 or more objects

When pointing at a window, activate it after 1 tenths of a second

Generate Script... Click to generate an automation script for the global testing options

Virtual Objects Limitations

- You can define Virtual Objects only for objects on which you can click or double-click and that record a Click or DblClick step.
- You can use Virtual Objects only when recording and running a test. You cannot insert any type of checkpoint on a Virtual Object, or use the Object Spy to view its properties.
- UFT does not support Virtual Objects for analog or low-level recording

Recovery Scenarios

What is a Recovery Scenario?

- Recovery Scenario is a mechanism by which UFT handles any unexpected windows, pop-ups or application crashes while the test is running so that the test is not interrupted.

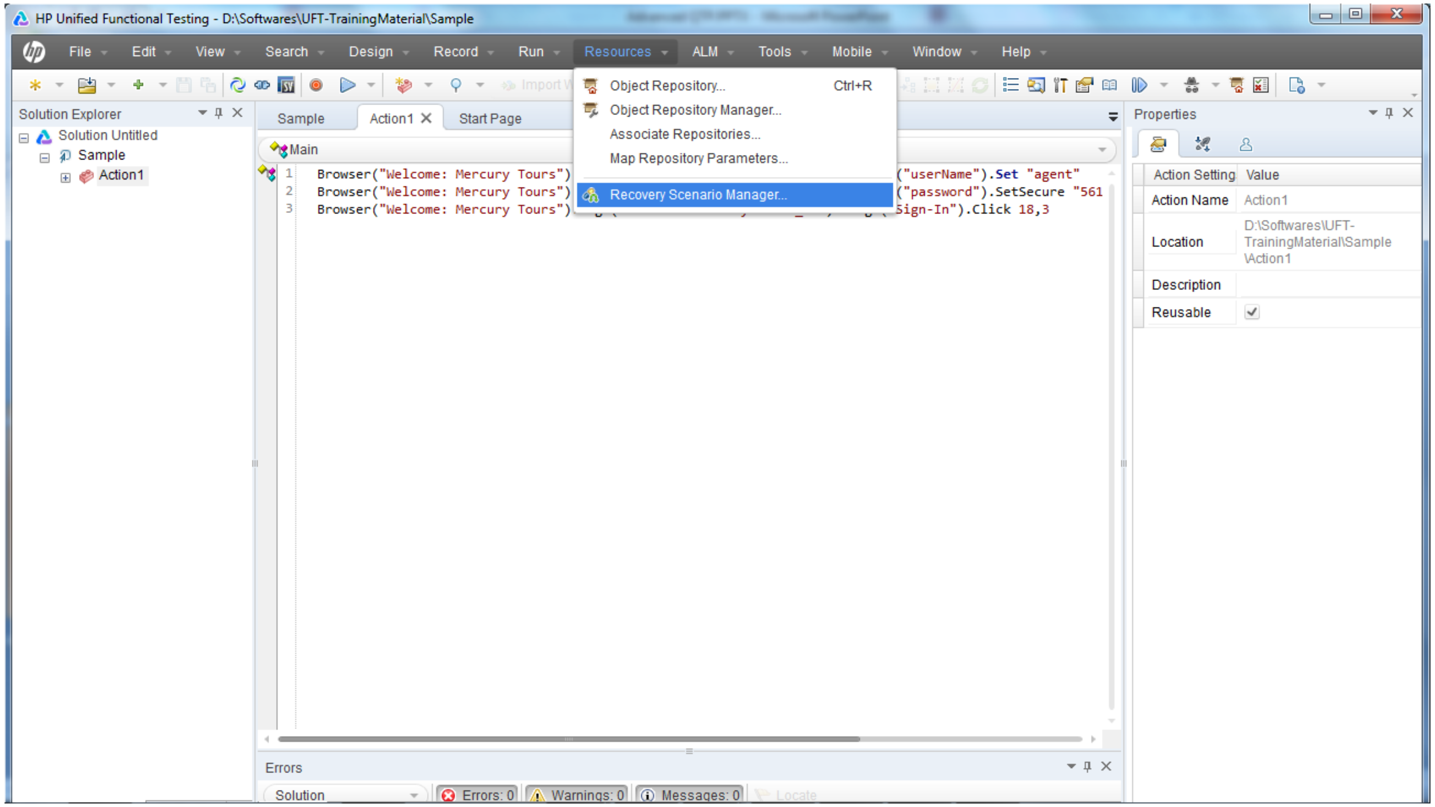
- Every type of recovery situation needs to be handled with a separate recovery scenario.

- The recovery scenarios continuously look for the recovery situations occurring in the application as long as the test is running.

- A Recovery scenario consists of 3 Stages
  - a) Trigger Event
  - b) Recovery Operation
  - c) Post Recovery Run Option

How to create a recovery scenario?

Select “Recovery Scenario Manager" from the "Resources" menu.

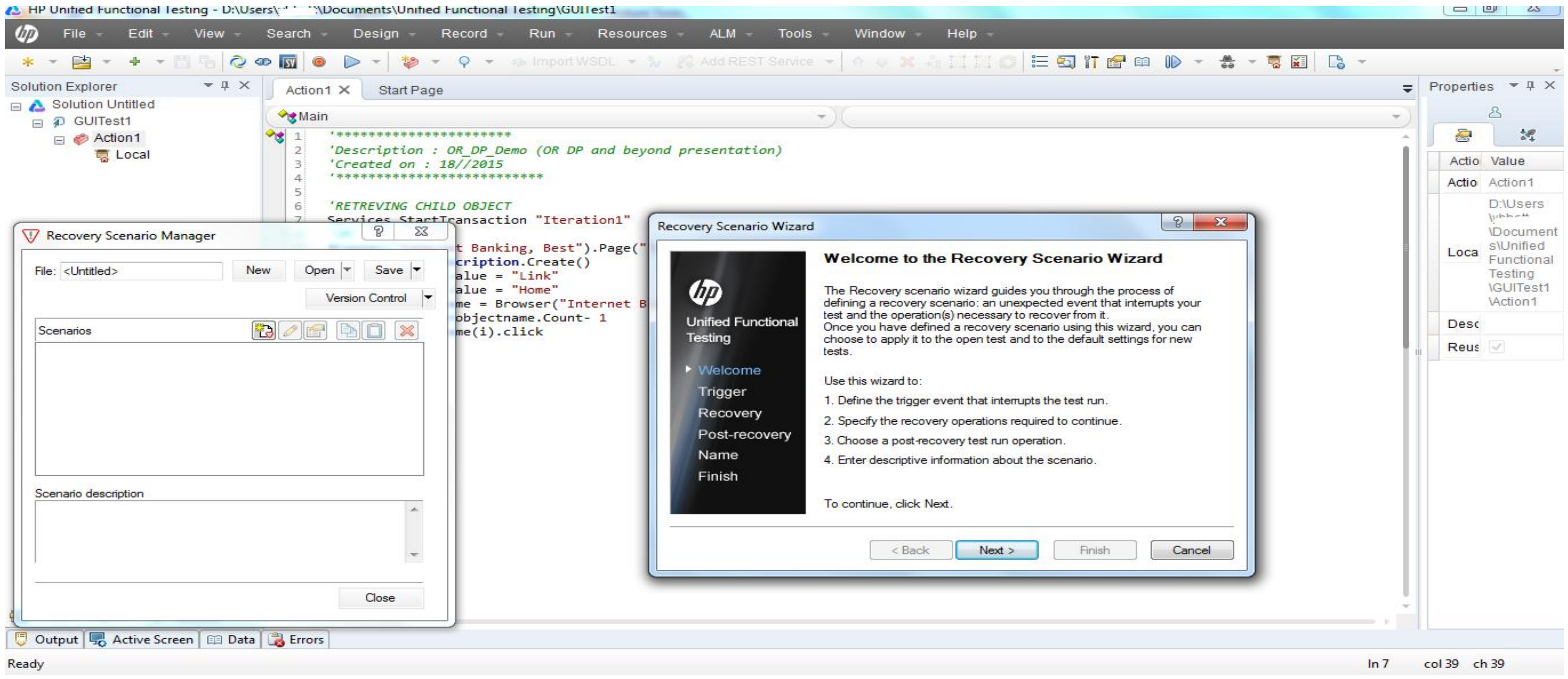

HP Unified Functional Testing - D:\Softwares\UFT-TrainingMaterial\Sample
File
Edit
View
Search
Design
Record
Run
Resources
ALM
Tools
Mobile
Window
Help
Object Repository... Ctrl+R
Object Repository Manager...
Associate Repositories...
Map Repository Parameters...
Recovery Scenario Manager...
Solution Explorer
Solution Untitled
Sample
Action1
Properties
Errors
Solution
Errors: 0
Warnings: 0
Messages: 0
Locate

```
Browser("Welcome: Mercury Tours")              ("userName").Set "agent"
Browser("Welcome: Mercury Tours")              ("password").SetSecure "561
Browser("Welcome: Mercury Tours")              Sign-In").Click 18,3
```

Action Setting	Value
Action Name	Action1
Location	D:\Softwares\UFT-TrainingMaterial\Sample \Action1
Description	
Reusable	✓

How to create a recovery scenario?

Recovery Scenario Wizards starts. Click Next on this screen.



HP Unified Functional Testing - D:\Users\ ...\Documents\Unified Functional Testing\GUITest1
File Edit View Search Design Record Run Resources ALM Tools Window Help
Import WSDL
Add REST Service
Solution Explorer
Solution Untitled
GUITest1
Action1
Local
Action1
Start Page
Main
Properties
Output
Active Screen
Data
Errors
Ready
ln 7
col 39
ch 39

```
'***********************
'Description : OR_DP_Demo (OR DP and beyond presentation)
'Created on : 18//2015
'**************************

'RETRIEVING CHILD OBJECT
Services.StartTransaction "Iteration1"
t Banking, Best").Page("
cription.Create()
alue = "Link"
alue = "Home"
me = Browser("Internet B
objectname.Count- 1
me(i).click
```

Actio	Value
Actio	Action1
Loca	D:\Users \...\Documents\Unified Functional Testing \GUITest1 \Action1
Desc	
Reus	✓

Recovery Scenario Manager

File: <Untitled>

New | Open | Save | Version Control

Scenarios

Scenario description

Close

Recovery Scenario Wizard

Unified Functional Testing

- Welcome
- Trigger
- Recovery
- Post-recovery
- Name
- Finish

Welcome to the Recovery Scenario Wizard

The Recovery scenario wizard guides you through the process of defining a recovery scenario: an unexpected event that interrupts your test and the operation(s) necessary to recover from it.
Once you have defined a recovery scenario using this wizard, you can choose to apply it to the open test and to the default settings for new tests.

Use this wizard to:

1. Define the trigger event that interrupts the test run.
2. Specify the recovery operations required to continue.
3. Choose a post-recovery test run operation.
4. Enter descriptive information about the scenario.

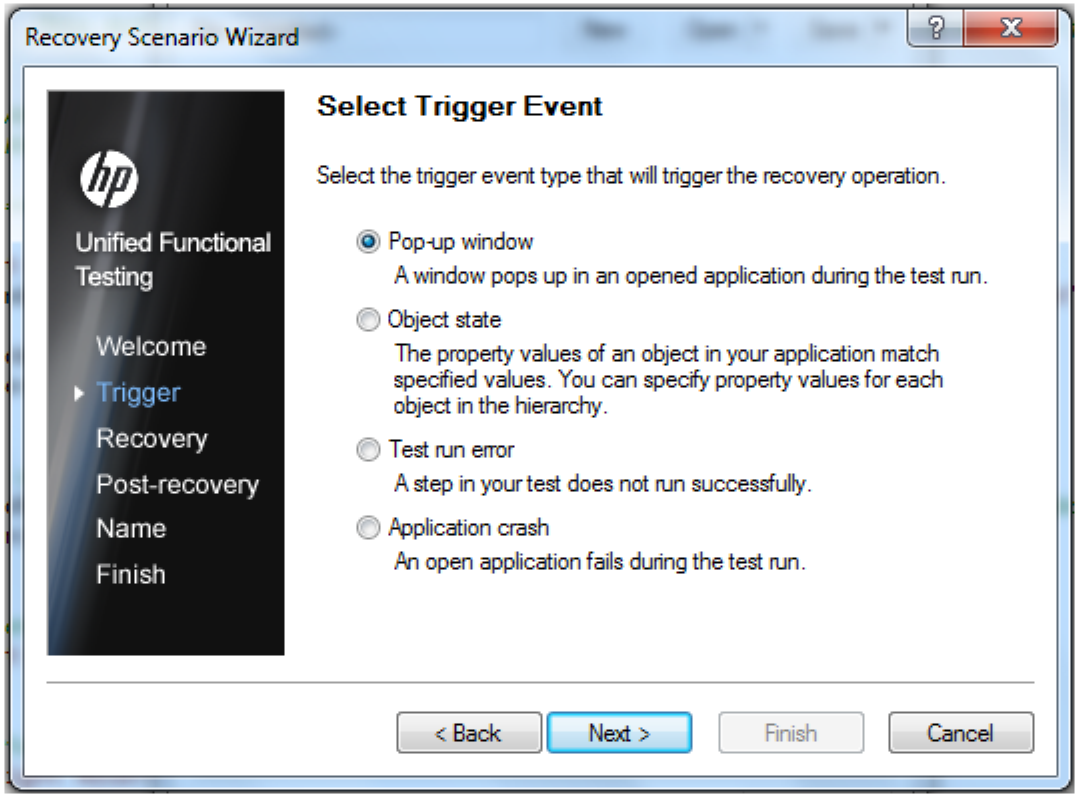
To continue, click Next.

< Back | Next > | Finish | Cancel

How to create a Recovery Scenario?

Depending on the desired type of recovery scenario, select the appropriate Radio button.

Recovery scenarios can be defined for unwanted Pop-up window, Object state, Test run error or application crash.



Select Trigger Event

Select the trigger event type that will trigger the recovery operation.

[✓ Pop-up window] A window pops up in an opened application during the test run.

Object state
The property values of an object in your application match specified values. You can specify property values for each object in the hierarchy.

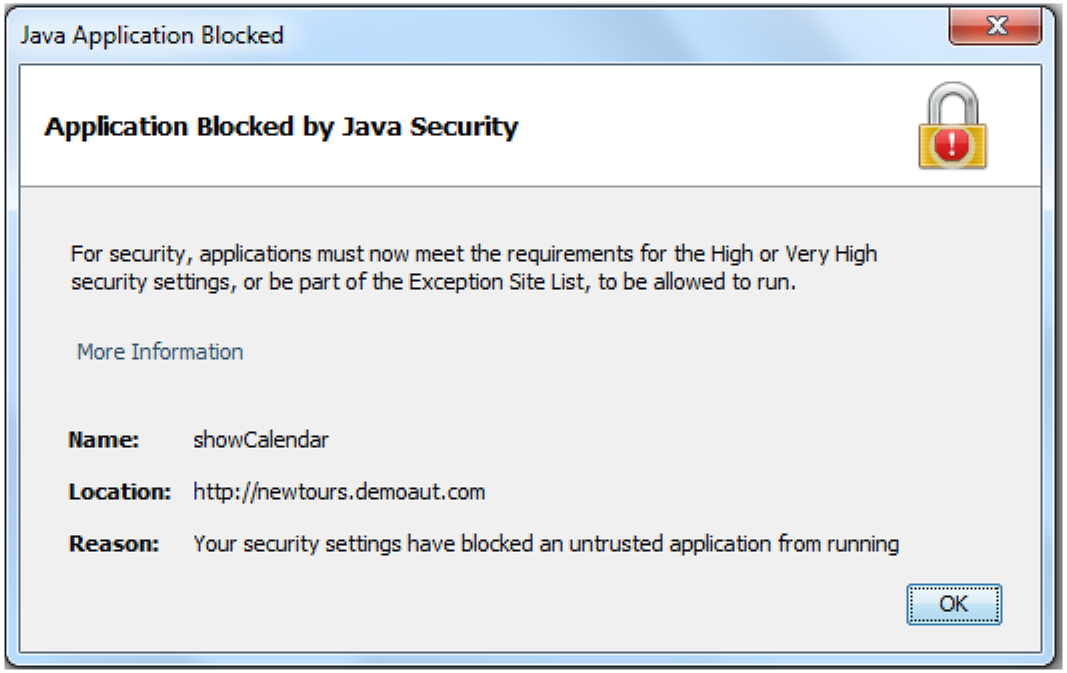
Test run error
A step in your test does not run successfully.

Application crash
An open application fails during the test run.

How to create a Recovery Scenario?

Click Next button to define Recovery Scenario for a pop-up window.

For Instance, following Pop-up window may appear while navigating from non-secure to a secure web page and will create Recovery Scenario for the Pop-up



Java Application Blocked

Application Blocked by Java Security

For security, applications must now meet the requirements for the High or Very High security settings, or be part of the Exception Site List, to be allowed to run.

More Information

Name: showCalendar

Location: http://newtours.demoaut.com

Reason: Your security settings have blocked an untrusted application from running

How to create a Recovery Scenario?

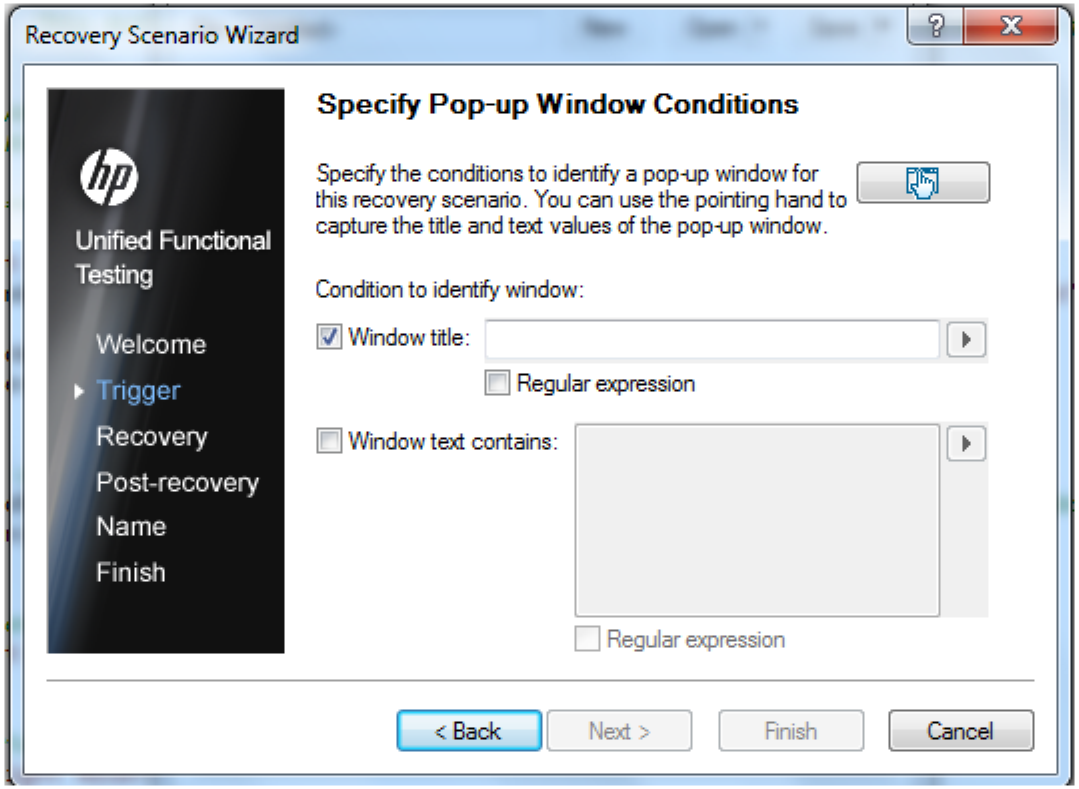
UFT captures the window title and window text.

Uncheck the checkbox "Window text contains".

Now UFT look for any security window with the title Security Information or any generic title.

If the window title changes dynamically with some pattern, Click the checkbox "Regular Expression" and provide the pattern.

Click Next


Recovery Scenario Wizard

Specify Pop-up Window Conditions

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.

Condition to identify window:

[✓] Window title:

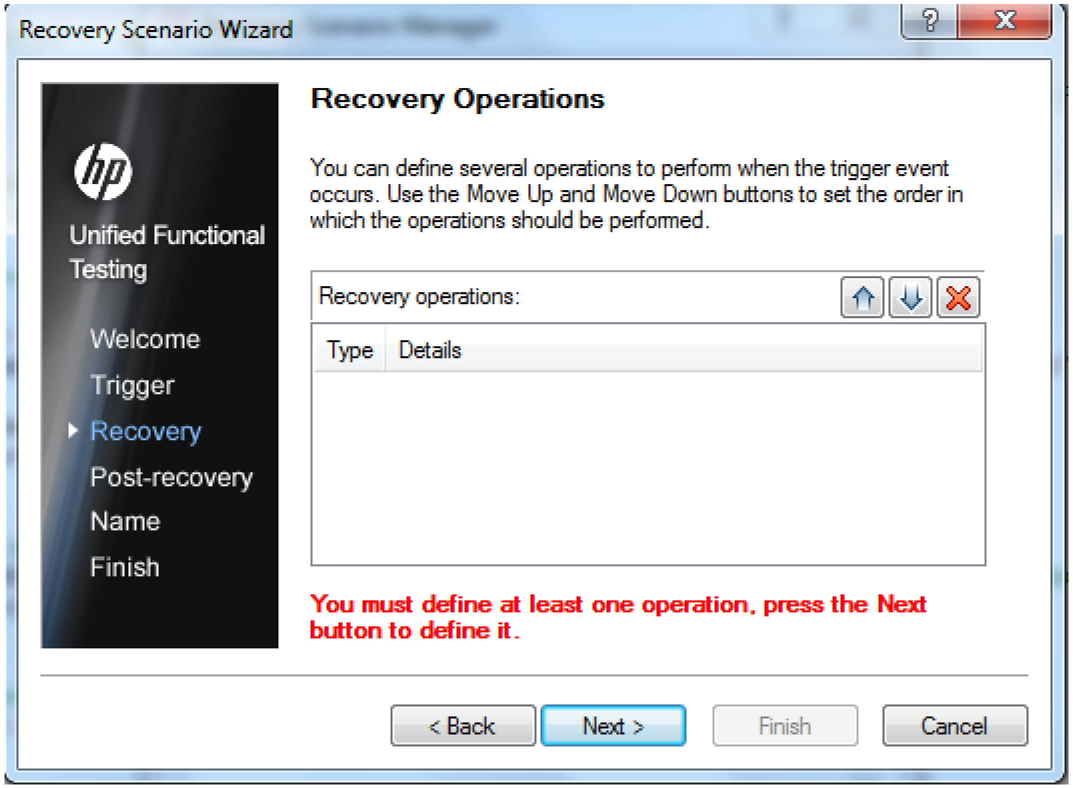
[ ] Regular expression

[ ] Window text contains:

[ ] Regular expression

How to create a Recovery Scenario?

Next screen informs us that we should define the recovery operation to be done in order to handle this window.



Recovery Scenario Wizard

Recovery Operations

You can define several operations to perform when the trigger event occurs. Use the Move Up and Move Down buttons to set the order in which the operations should be performed.

Recovery operations:

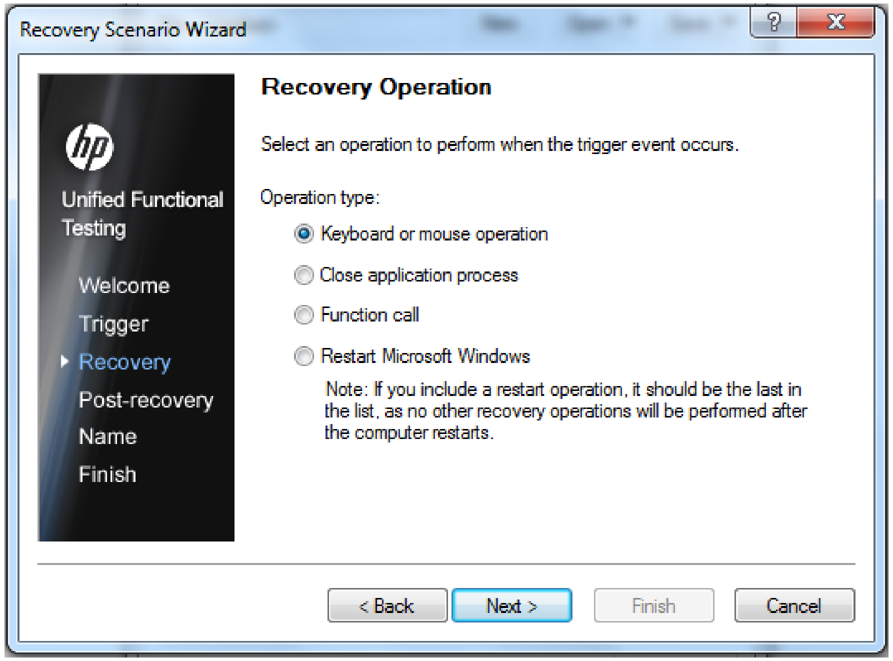
Type	Details

You must define at least one operation, press the Next button to define it.

How to create a Recovery Scenario?

Click Next button to view the Recovery Operation window.

Select the appropriate action to be performed.



Recovery Scenario Wizard

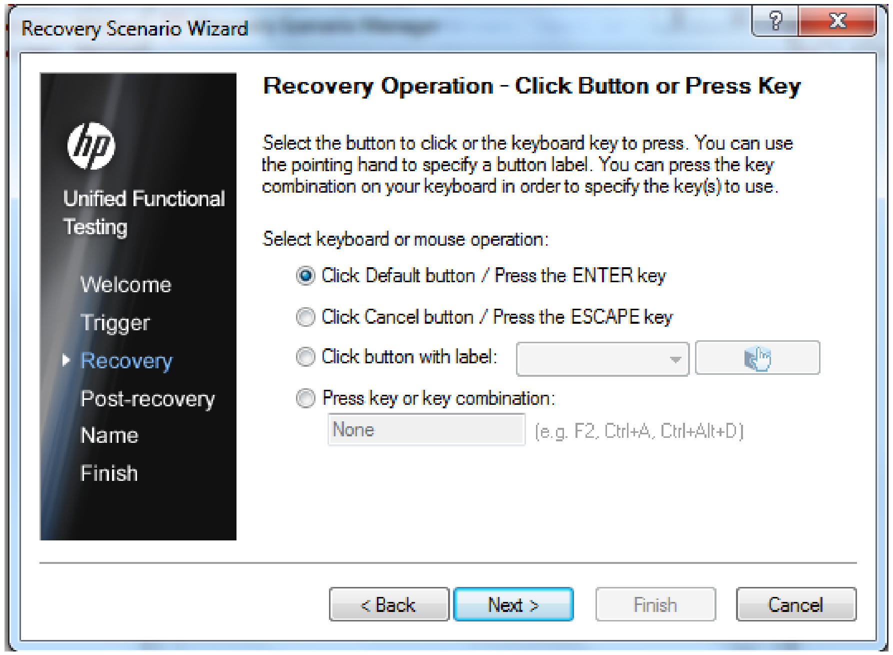
Recovery Operation

Select an operation to perform when the trigger event occurs.

Operation type:

- [✓ Keyboard or mouse operation]
- Close application process
- Function call
- Restart Microsoft Windows

Note: If you include a restart operation, it should be the last in the list, as no other recovery operations will be performed after the computer restarts.



Recovery Scenario Wizard

Recovery Operation - Click Button or Press Key

Select the button to click or the keyboard key to press. You can use the pointing hand to specify a button label. You can press the key combination on your keyboard in order to specify the key(s) to use.

Select keyboard or mouse operation:

- [✓ Click Default button / Press the ENTER key]
- Click Cancel button / Press the ESCAPE key
- Click button with label:
- Press key or key combination:

None (e.g. F2, Ctrl+A, Ctrl+Alt+D)

How to create a Recovery Scenario?

• Keyboard or mouse operation allows us to click a button on the screen.

• Close application process allows us to kill the process which starts the unwanted window so we can continue with testing.

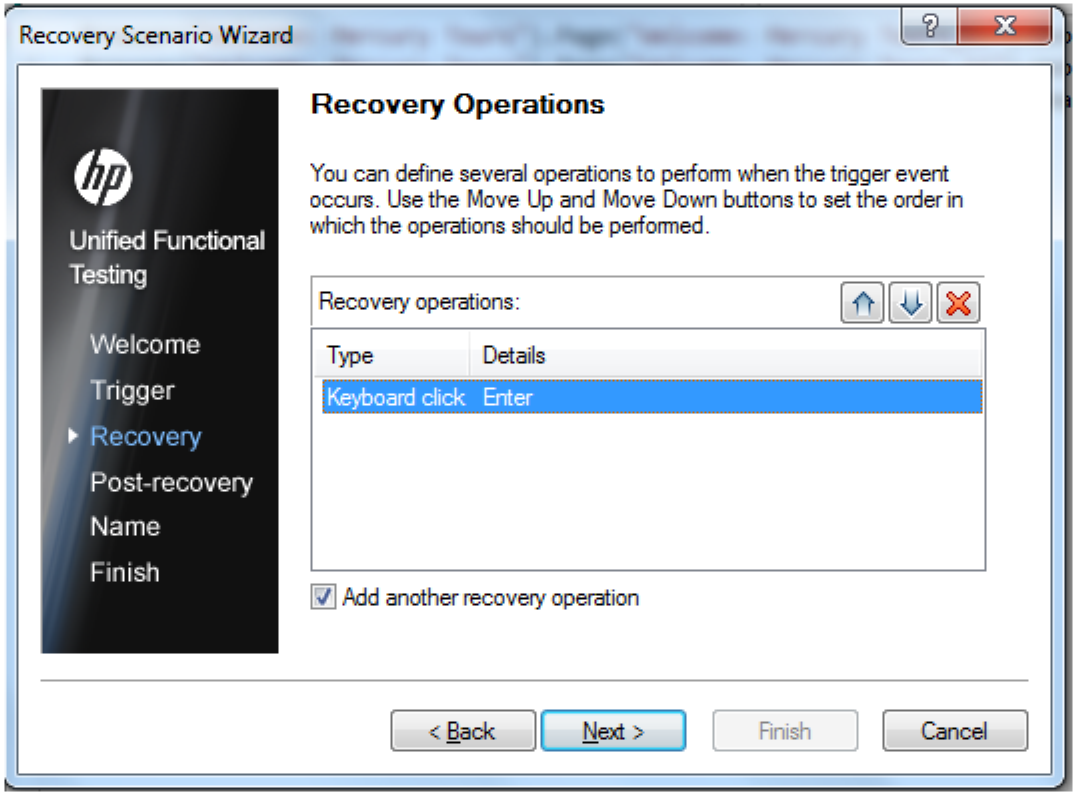
• Function call will allow us to write a user-defined function to handle the unwanted window.

• Restart Microsoft Windows allows us to restart the windows all together if needed.

In above example, click on the first radio button "Keyboard or mouse operation".

How to create a Recovery Scenario?

We can show the button we want UFT to click using the hand icon in the following screen and click Next button.


Recovery Scenario Wizard

Recovery Operations

You can define several operations to perform when the trigger event occurs. Use the Move Up and Move Down buttons to set the order in which the operations should be performed.

Recovery operations:

Type	Details
Keyboard click	Enter

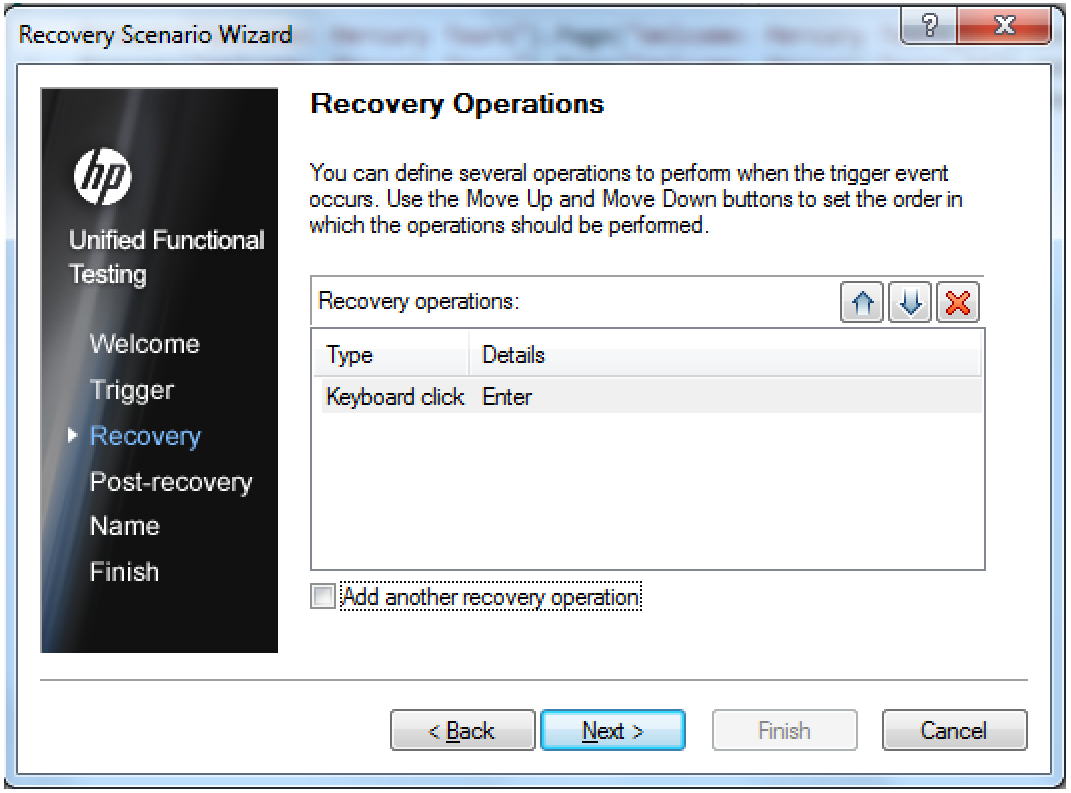
☑ Add another recovery operation

How to create a Recovery Scenario?

We can add another recovery scenario if needed from the following screen.

If we don't want to create another scenario uncheck the checkbox "Add another recovery operation"

Click "Next"



Recovery Operations

You can define several operations to perform when the trigger event occurs. Use the Move Up and Move Down buttons to set the order in which the operations should be performed.

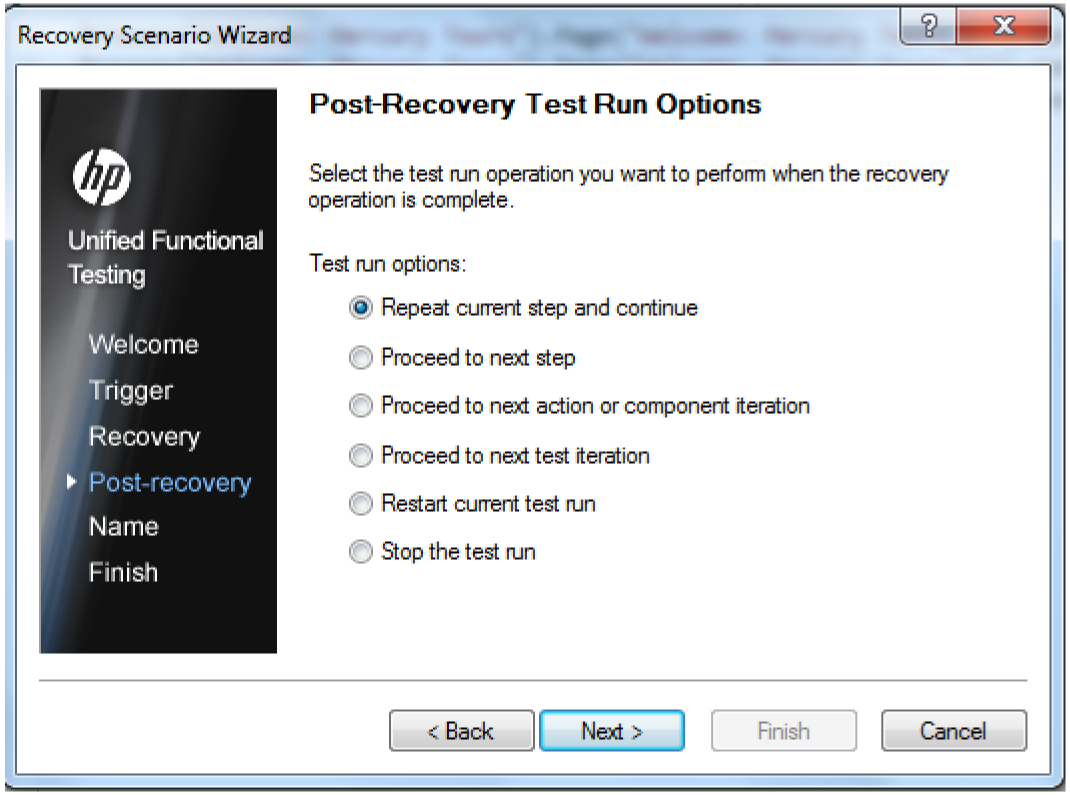
Recovery operations:

Type	Details
Keyboard click	Enter

☐ Add another recovery operation

How to create a Recovery Scenario?

Following screen allows us to define the Post-recovery operation to be performed.



Post-Recovery Test Run Options

Select the test run operation you want to perform when the recovery operation is complete.

Test run options:

- [✓ Repeat current step and continue]
- Proceed to next step
- Proceed to next action or component iteration
- Proceed to next test iteration
- Restart current test run
- Stop the test run

How to create a Recovery Scenario?

Since the recovery scenario is kicked off only when the error is about to be thrown since UFT could not find an object because of the unwanted window or object state it is logical to re-execute that particular statement again after the post-recovery operation.

Hence select the option "Repeat current step and continue".

If the situation is different, make the appropriate selection depending on the desired operation.

If the exception does not allow us to test the application anymore in this test run, select the last radio button "Stop the test run"

How to create a Recovery Scenario?

• Use “Proceed to next step” if you want to continue with the test.

• Use “Proceed to next action” if you want to skip the current action in the flow and continue with the next action.

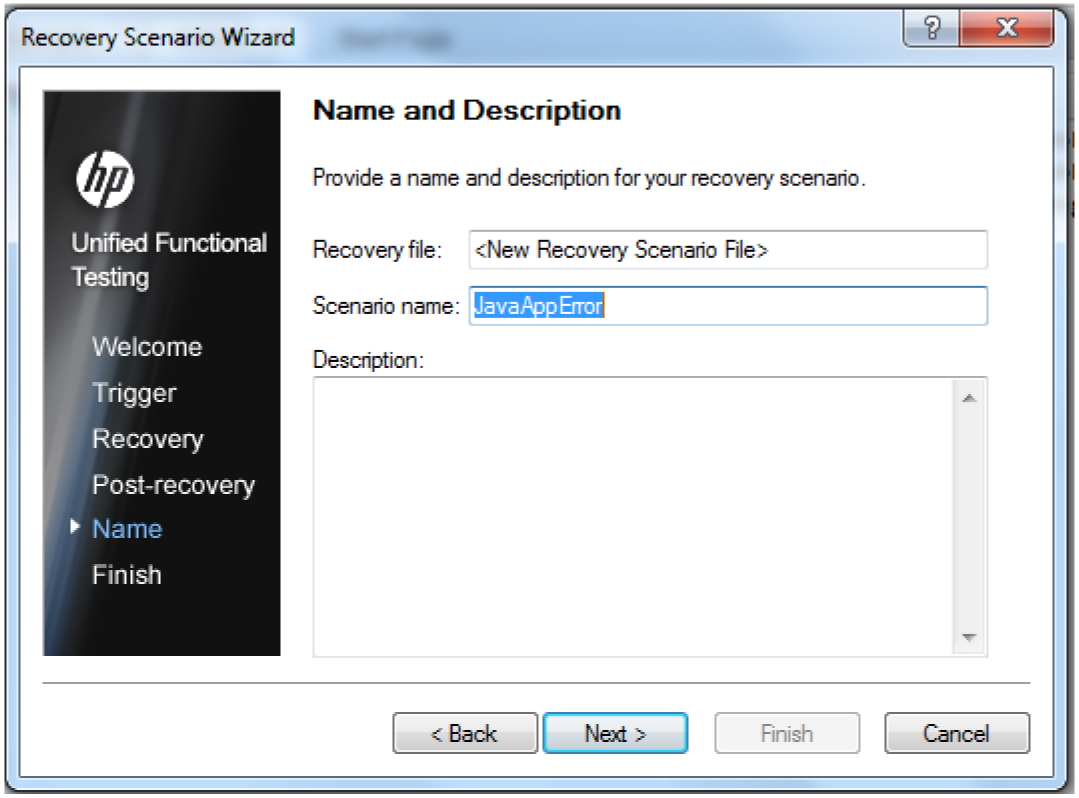
• Use “Proceed to next test iteration” if you want to skip the current row of global row and continue with the next row of Global sheet.

• Use “Restart current test run” if you want to start the test altogether.

Click Next.

How to create a Recovery Scenario?

Provide a name for the Recovery Scenario and click "Next".



Recovery Scenario Wizard

Name and Description

Provide a name and description for your recovery scenario.

Recovery file: <New Recovery Scenario File>

Scenario name: JavaAppError

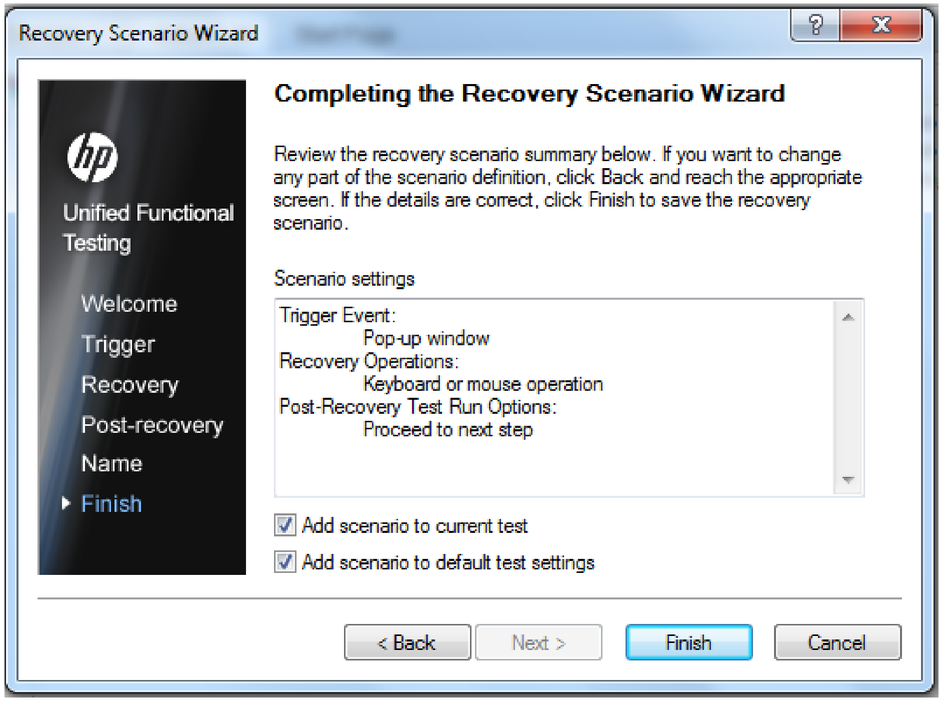
Description:

How to create a Recovery Scenario?

Make sure that the checkboxes "Add scenario to current test" and "Add scenario to default test settings" are selected as shown below.

This will add the Recovery Scenario to the current and the UFT settings as well.

Click "Finish" and follow the further steps to save the recovery scenario to a recovery file



Completing the Recovery Scenario Wizard

Review the recovery scenario summary below. If you want to change any part of the scenario definition, click Back and reach the appropriate screen. If the details are correct, click Finish to save the recovery scenario.

Scenario settings

```
Trigger Event:
        Pop-up window
Recovery Operations:
        Keyboard or mouse operation
Post-Recovery Test Run Options:
        Proceed to next step
```

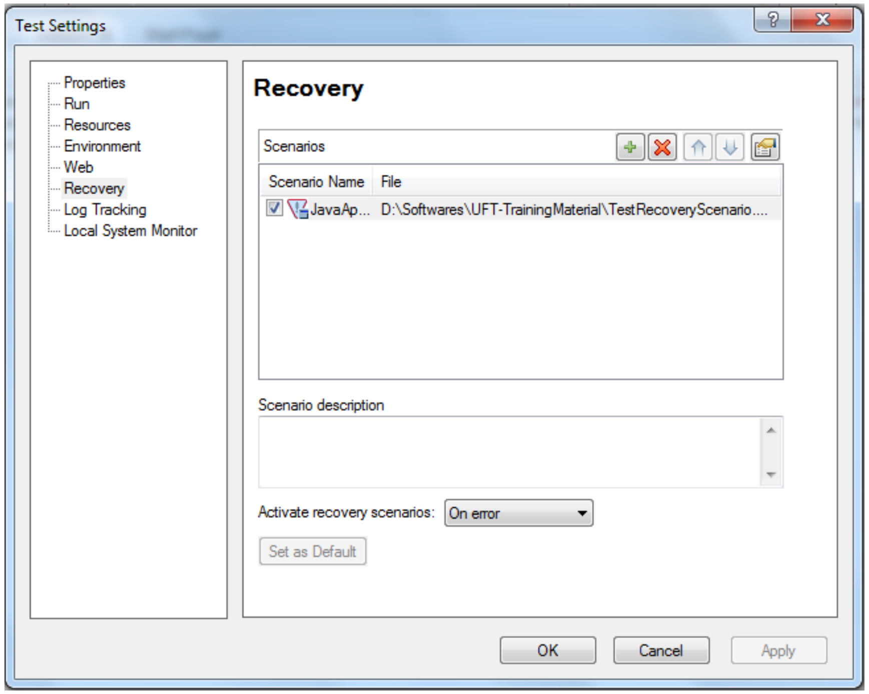
☑ Add scenario to current test

☑ Add scenario to default test settings

Associate Recovery Scenario

To enable/disable specific recovery scenarios:

- Select the check box to the left of one or more individual scenarios to enable them.
- Clear the check box to the left of one or more individual scenarios to disable them.



Test Settings

- Properties
- Run
- Resources
- Environment
- Web
- Recovery
- Log Tracking
- Local System Monitor

Recovery

Scenarios

Scenario Name	File
✓ JavaAp...	D:\Softwares\UFT-TrainingMaterial\TestRecoveryScenario....

Scenario description

Activate recovery scenarios: On error

ALM Integration

Prerequisites to connect UFT with ALM

- Check Allow other HP products to run tests and components present under Tools > Options > Run in UFT

- If you are running the tests on the same computer where you have ALM client installed, then you will need:
  - UFT Connectivity Add-In
  - UFT Add-in

- If you are running the tests on the different computer than where you have QC/ALM client installed, then you will need:
  - UFT Add-in where ALM client is installed.
  - UFT Add-in and ALM connectivity Add-in where UFT is installed.

- QC connectivity can be found at ALM server URL > 'Add-Ins Page' link > 'ALM Connectivity' link > 'Download Add-in'

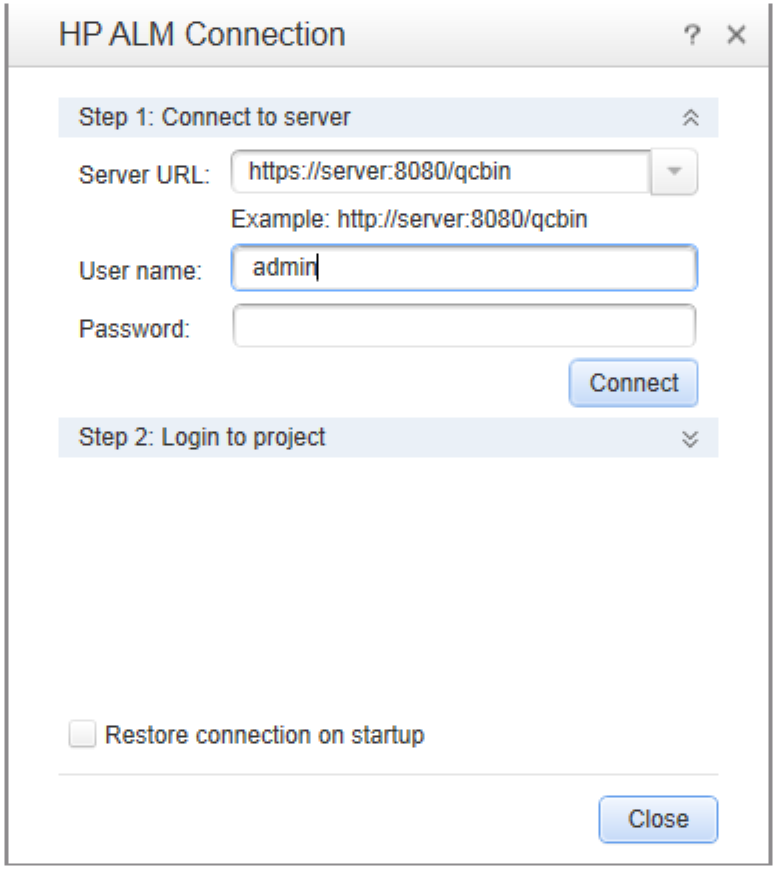
- UFT Add-in can be found at ALM server URL > 'Add-Ins Page' link > 'More ALM Add-ins' link > Download and install UFT Add-in according to its version

Connecting UFT to ALM

- Start UFT and from File option select ALM.

- In the server connection area in server text box enter http://<machine name>/<qcbin>and click on connect

- In Project Connection area, click on connect after selecting project


HP ALM Connection

Step 1: Connect to server

Server URL: https://server:8080/qcbin

Example: http://server:8080/qcbin

User name: admin

Password:

Enabling ALM to Run Tests on a UFT Computer

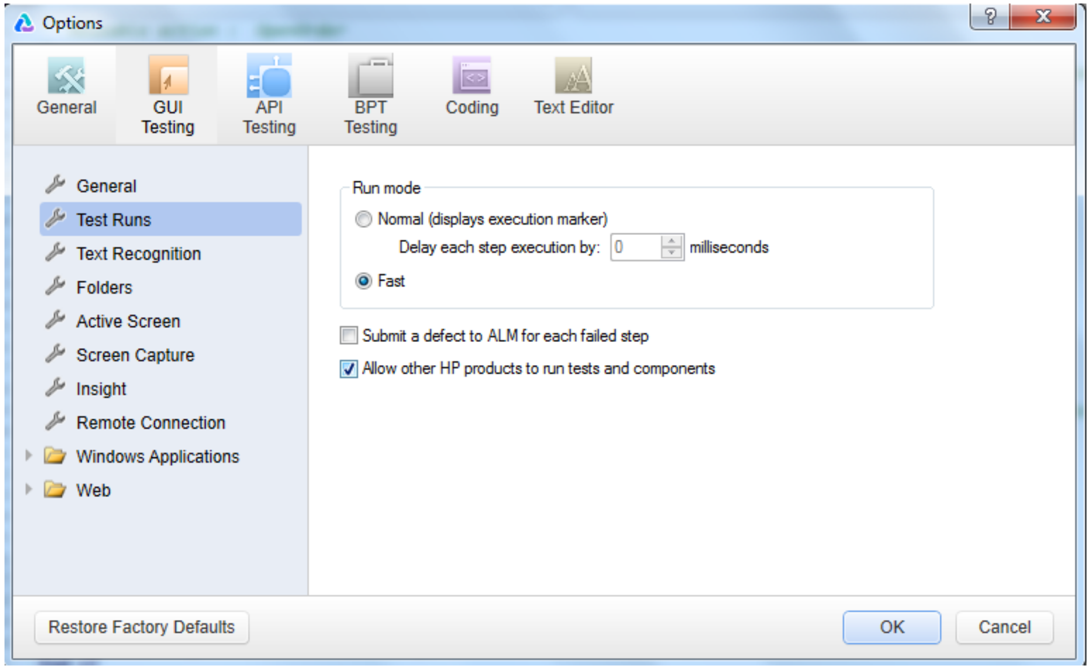
To enable remote Quality Center clients to run tests on Your Quick Test computer:

• Open UFT.

• Choose Tools > Options or click the Options toolbar button. The Options dialog box opens.

• Click the Run tab.

• Select the "Allow other HP products to run tests and components" check box.



Run mode

( ) Normal (displays execution marker)

Delay each step execution by: 0 milliseconds

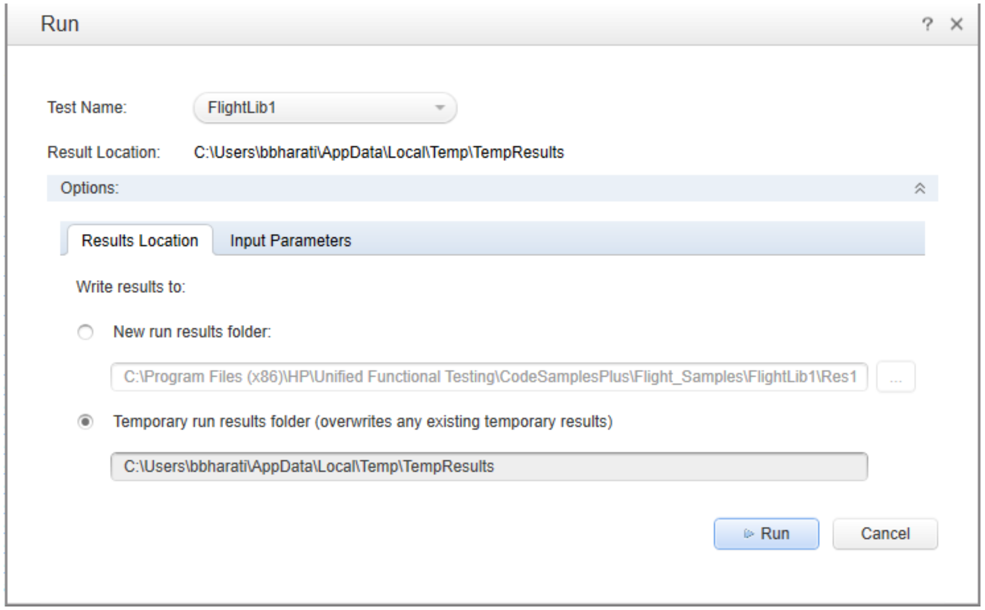
(●) Fast

☐ Submit a defect to ALM for each failed step

☑ Allow other HP products to run tests and components

Running a Test Stored in a Quality Center Project

- Test can be executed either from ALM Test lab or from UFT
- To execute the Test from UFT
  - Click on Test->Run
  - To save the run results, you specify a name for the run session and a test set in which to store the results.



Test Name: FlightLib1

Result Location: C:\Users\bbharati\AppData\Local\Temp\TempResults

Write results to:

( ) New run results folder:

C:\Program Files (x86)\HP\Unified Functional Testing\CodeSamplesPlus\Flight_Samples\FlightLib1\Res1

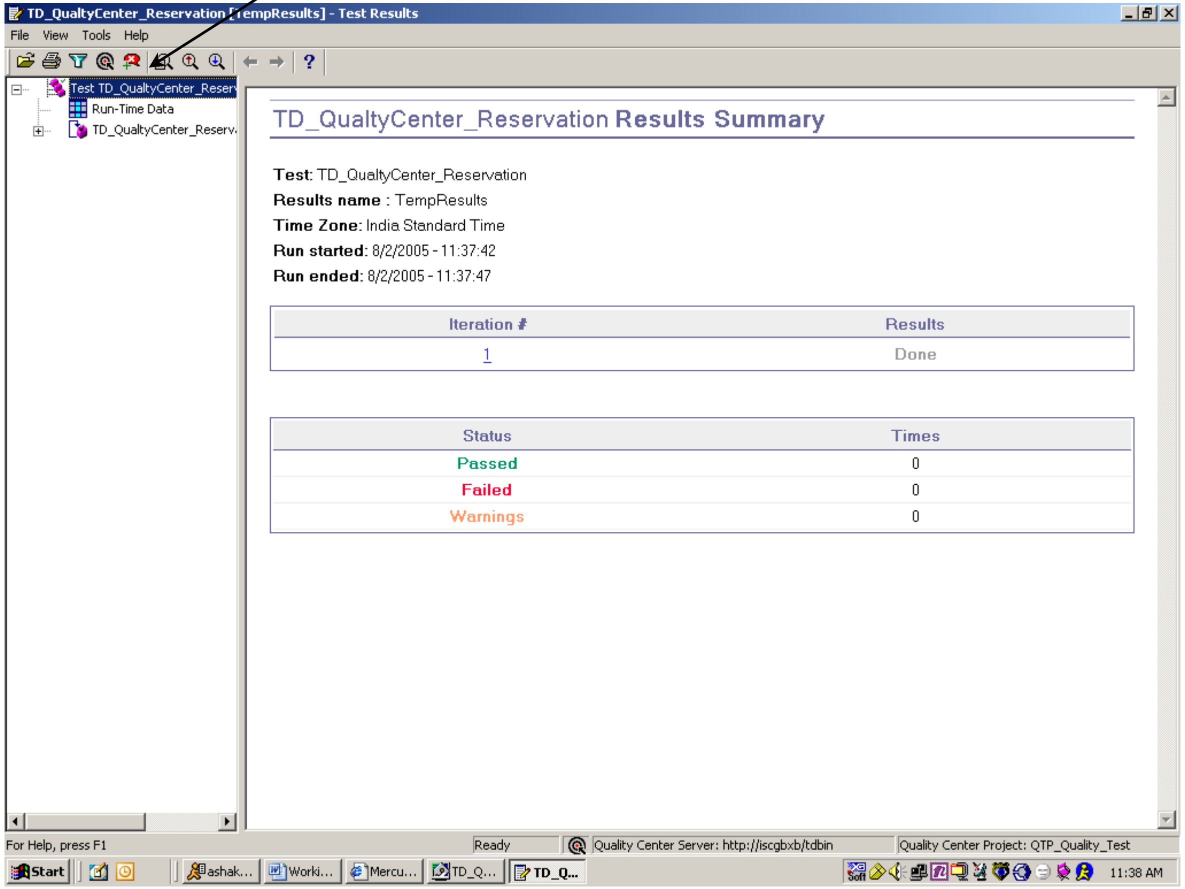
(●) Temporary run results folder (overwrites any existing temporary results)

C:\Users\bbharati\AppData\Local\Temp\TempResults

Submitting Defects During a Run Session

- Run the test and result window would displayed
- Select add defect option given in the tool bar just near to quality center icon a window get open as shown below

Defect Option


TD_QualityCenter_Reservation [TempResults] - Test Results
File View Tools Help
Test TD_QualtyCenter_Reser
Run-Time Data
TD_QualtyCenter_Reserv.
For Help, press F1
Ready
Quality Center Server: http://iscgbxb/tdbin
Quality Center Project: QTP_Quality_Test
Start
11:38 AM

TD_QualtyCenter_Reservation **Results Summary**

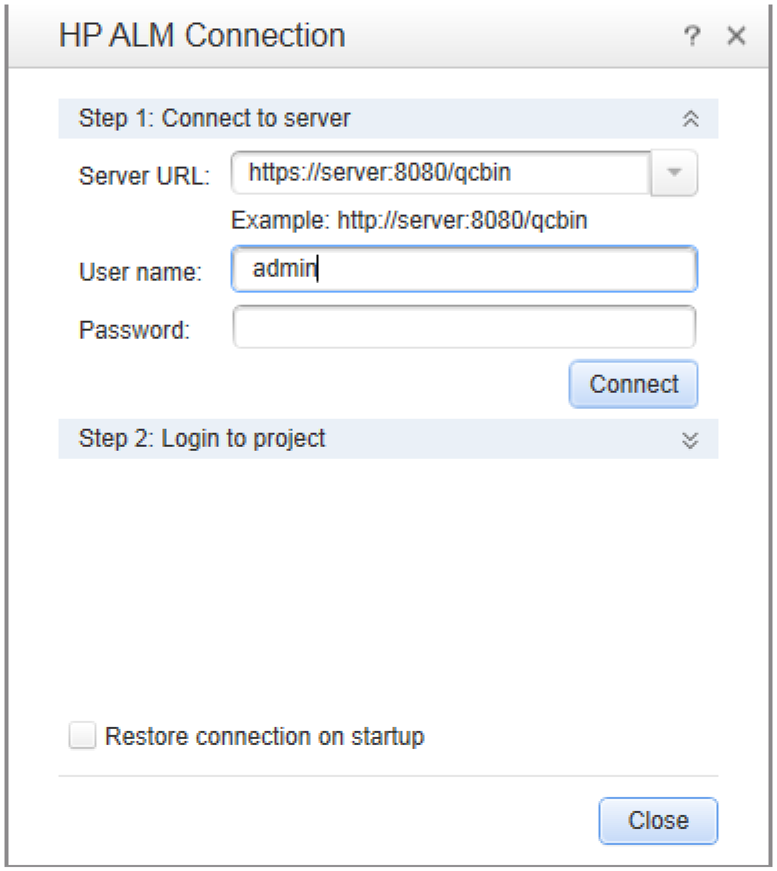
Test: TD_QualtyCenter_Reservation
Results name : TempResults
Time Zone: India Standard Time
Run started: 8/2/2005 - 11:37:42
Run ended: 8/2/2005 - 11:37:47

Iteration #	Results
1	Done

Status	Times
Passed	0
Failed	0
Warnings	0

Disconnecting UFT to ALM

- Select File > HP ALM Connection or click the Quality Center Connection toolbar button.
- In project connection area click on Disconnect.
- In server connection area click on Disconnect.



HP ALM Connection

Step 1: Connect to server

Server URL: https://server:8080/qcbin

Example: http://server:8080/qcbin

User name: admin

Password:

Step 2: Login to project

Restore connection on startup

Thank You ☺