

Advanced Features of QTP

Contents

Script Debugging

Object Repository

Descriptive Programming

Regular Expression

Virtual Objects

Recovery Scenarios

Quality Center Integration

Script Debugging

Script debugging

- **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 QTP scripts.**
- **For starting the debugging process, go to "Debug" 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.
- **Variable Tab:**
Displays the current values and types of all variables in the main script of the current action, or in a selected subroutine, and enables you to modify their values.
- **Command Tab:**
Enables you to run VBScript commands in your paused run session

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, QTP 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.

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:

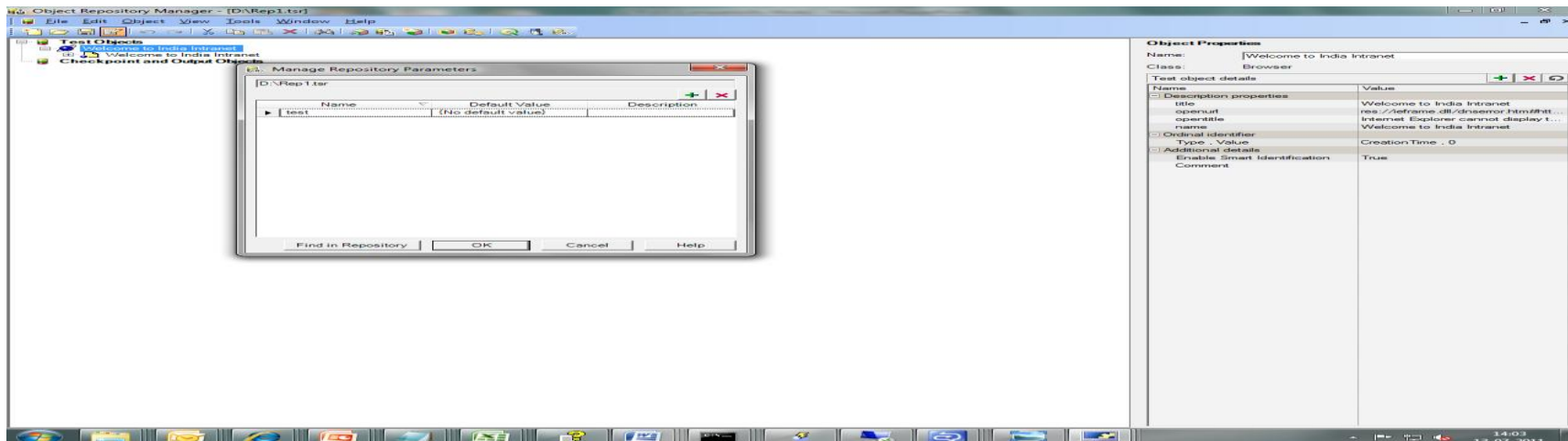
- **Index:** Indicates the order in which the object appears in the application code relative to other objects with an otherwise identical description.
- **Location:** Indicates the order in which the object appears within the parent window, frame, or dialog box relative to other objects with an otherwise 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 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.
- 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

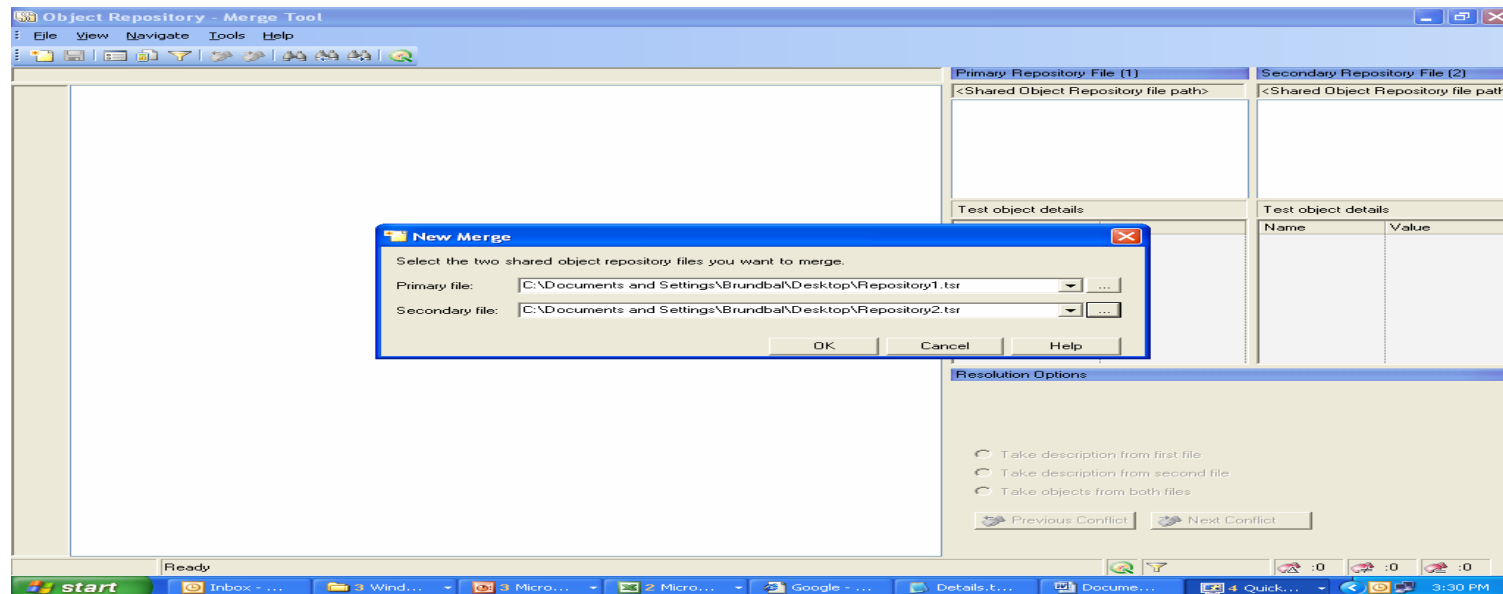
Managing Repository Parameters

- 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 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 Comparison Tool

- **QuickTest 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**

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?

- **Programmatic description to instruct QTP 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:

- **Dynamic object properties**
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:**

1

Throw the properties and values straight into a command

First Method...

```
1: VB'Window("Property1:=Value1").Click
```

**It's the good old syntax you know, except the string between the ()
is not the OR name.**

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 😊

VBWindow(“height:=400”, “title:=New Document”).Maximize

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...

```
1: Set oDesc = Description.Create
2:
3: oDesc("Property1").Value = "Value1"
4:
5: VBWindow(oDesc).Maximize
```

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

oDesc("Property1").RegularExpression = False

Descriptive Programming...

You can use DP with OR as shown below

VBWindow("OR").VBButton("text:=OK").Click

Or (when oDesc is a description object):

VBWindow("OR").VBButton(oDesc).Click

Descriptive Programming...

But, you can only start from OR, and move to DP

So this will not work:

VBWindow("title:=notgood").VBButton("clickme").Click

Descriptive Programming...

-
- You can store the objects in collections.
- When QTP finds two object which match the same description, it freezes 😞

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

www.AdvancedQTP.com - Descriptive Pro... Please fill all text fields before proceeding

☐ CRM

☐ 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... Please fill all text fields before proceeding

☐ CRM

☐ Web - Hosting Name

☐ Enterprise App. E-Mail

☐ Web - Email App. Comments

☐ Enterprise DB

☐ Math App. Submit

Example by www.AdvancedQTP.com

Run Error



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.

Stop

Retry

Skip

Debug

Help

Details >>

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

www.AdvancedQTP.com - Descriptive Pro... Please fill all text fields before proceeding

☐ CRM ☐ Web - Hosting ☐ Enterprise App. ☐ Web - Email App. ☐ Enterprise DB ☐ Math App.

Name

E-Mail

Comments

Submit

Example by www.AdvancedQTP.com

www.AdvancedQTP.com - Descriptive Pro... Please fill all text fields before proceeding

☐ CRM ☐ Web - Hosting ☐ Enterprise App. ☐ Web - Email App. ☐ Enterprise DB ☐ Math App.

Name

E-Mail

Comments

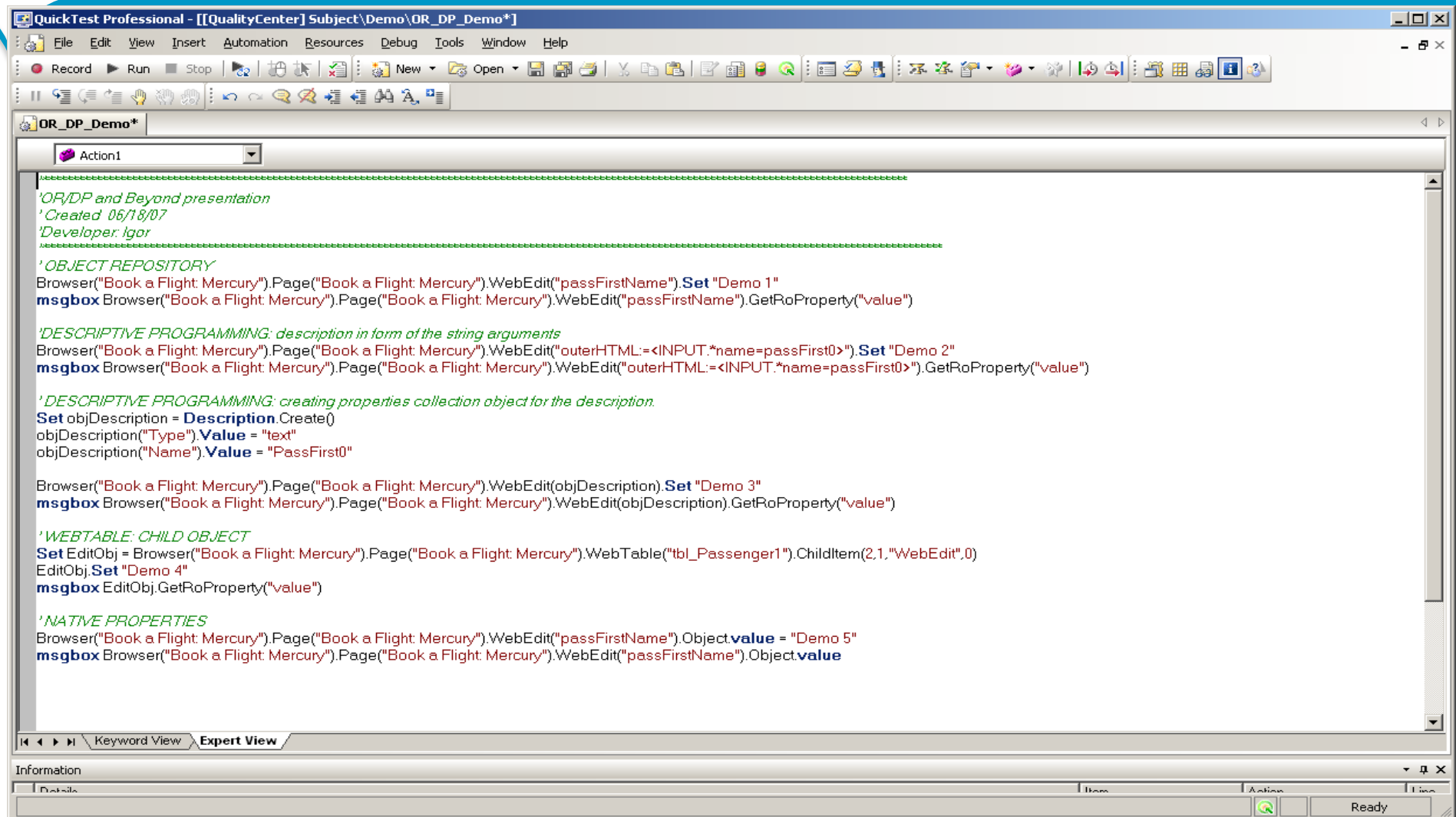
Submit

Example by www.AdvancedQTP.com

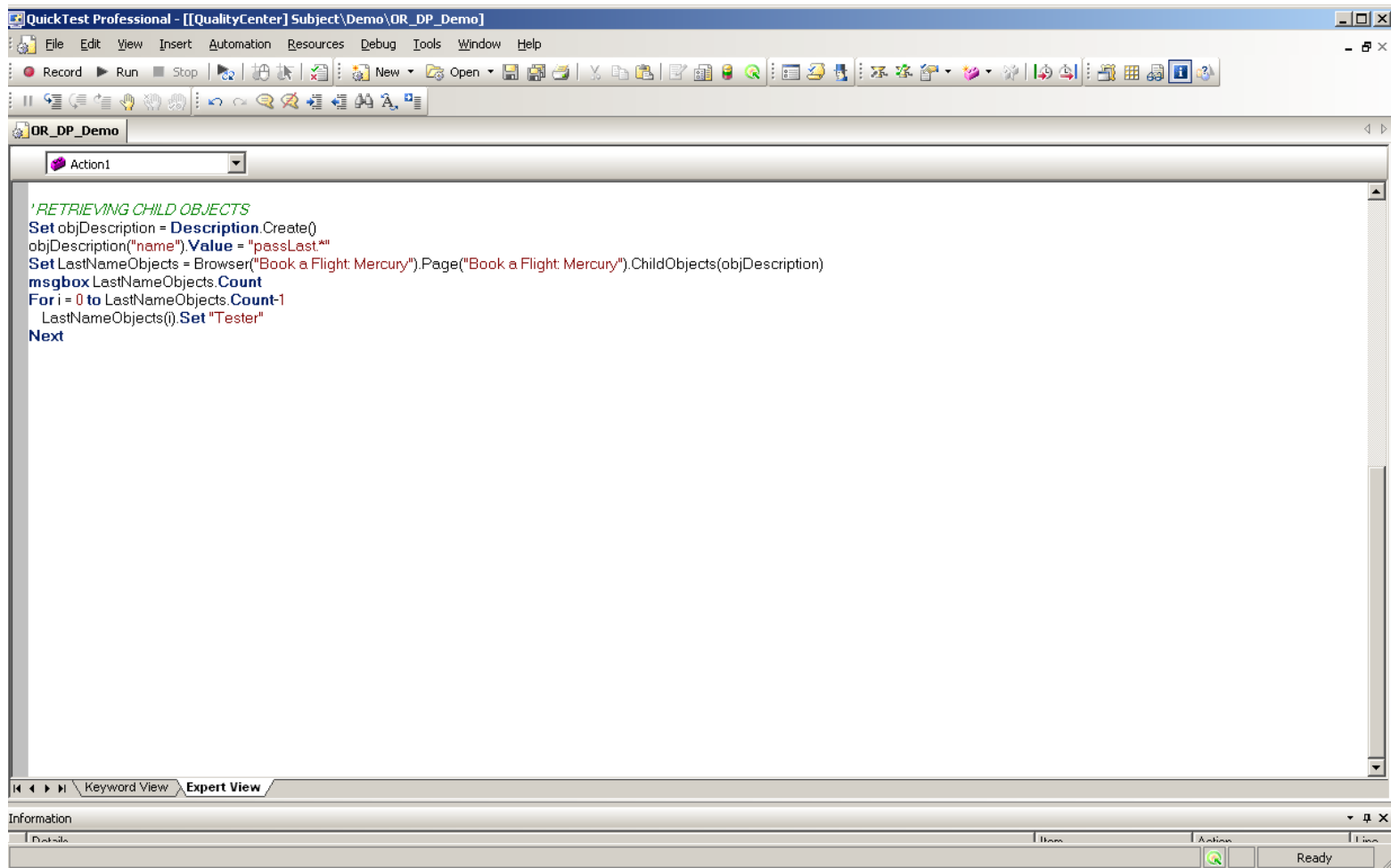
DP has a magic property: “index”, which allows us to tell the double objects apart

Index is a zero-based counter

Different ways to work with objects



ChildObjects method – using Collection Object



ChildObjects method – using Collection Object

Last Name fields
are populated

The screenshot shows a web browser window titled "Book a Flight: Mercury Tours - Microsoft Internet Explorer". The address bar shows "http://newtours.demoaut.com/mercurypurchase.php". The page displays flight details for "Blue Skies Airlines" with two flights: 360 and 630, both from Acapulco to Acapulco, in Coach class for \$270 each. The total price including taxes is \$1169. The "Passengers" section shows two passengers, both with "Demo 5" as the first name and "Tester" as the last name. The "Credit Card" section shows an American Express card. The "Billing Address" section shows "1325 Borregas Ave." and "Sunnyvale".

FLIGHT	CLASS	PRICE
Blue Skies Airlines 360	Coach	270
Acapulco to Acapulco		11/6/2007
FLIGHT	CLASS	PRICE
Blue Skies Airlines 630	Coach	270

Passengers: 2
Taxes: \$89
Total Price (including taxes): \$1169

Passengers

First Name:	Last Name:	Meal:
Demo 5	Tester	No preference
	Tester	No preference

Credit Card

Card Type:	Number:	Expiration:
American Express		None None

First Name: Middle: Last:

Billing Address

Address: 1325 Borregas Ave.
City: Sunnyvale

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 QTP 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.

Use Of Regular Expressions

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

- Use Regular Expressions only for values of type string.
- Use Regular Expressions for Property Values
- Using Regular Expressions for Checkpoints

For example, if you want to create a text checkpoint on a date text string, but the displayed date changes according to the current date. If we define the date as a regular expression, the checkpoint checks that the captured text string matches the expected date format, rather than checking the exact date value.

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

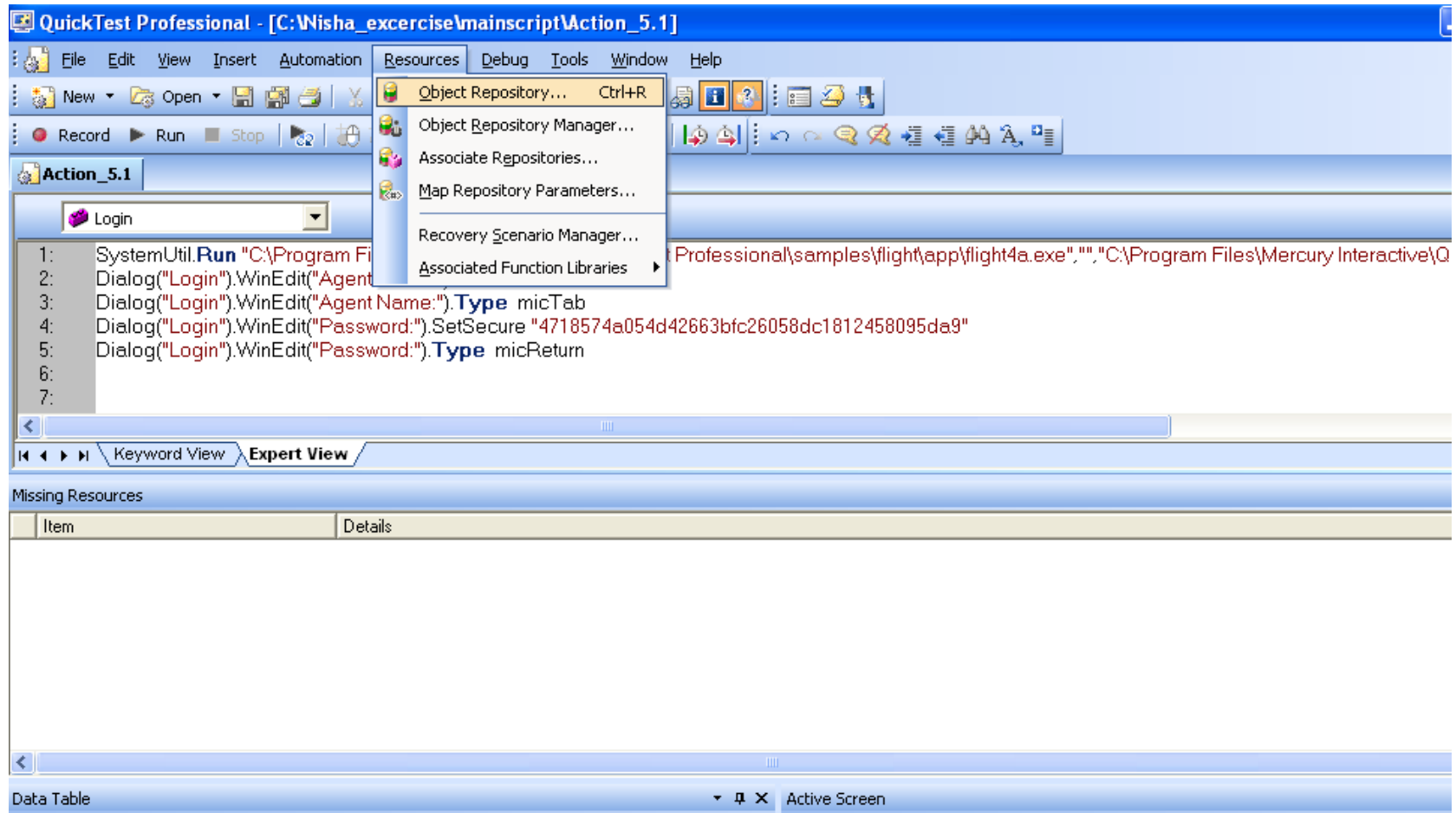
Fax Name:		Order:	Flight:	Date:
<input type="text" value="sdfsda"/>		<input type="text" value="15"/>	<input type="text" value="15801"/>	<input type="text" value="12/12/12"/>
From:	Departure:	To:	Arrival:	
<input type="text" value="Denver"/>	<input type="text" value="10:24 AM"/>	<input type="text" value="Frankfurt"/>	<input type="text" value="11:09 AM"/>	
Class:	# Tickets:	Ticket Price:	Total:	
<input type="text" value="Economy"/>	<input type="text" value="1"/>	<input type="text" value="104.20"/>	<input type="text" value="104.20"/>	
Fax Number:		Agent Signature:		
<input type="text" value="1.1.1.1"/>		<input type="text"/>		
<input type="checkbox"/> Send Signature with order				
<input type="button" value="Preview Fax"/>		<input type="button" value="Send"/>		<input type="button" value="Cancel"/>
<input type="button" value="Clear Signature"/>				

During Playback
Fax Order No. 16

Fax Name:		Order:	Flight:	Date:
<input type="text" value="dfdfdf"/>		<input type="text" value="16"/>	<input type="text" value="15805"/>	<input type="text" value="12/12/12"/>
From:	Departure:	To:	Arrival:	
<input type="text" value="Denver"/>	<input type="text" value="12:48 PM"/>	<input type="text" value="Frankfurt"/>	<input type="text" value="01:33 PM"/>	
Class:	# Tickets:	Ticket Price:	Total:	
<input type="text" value="Economy"/>	<input type="text" value="1"/>	<input type="text" value="100.30"/>	<input type="text" value="100.30"/>	
Fax Number:		Agent Signature:		
<input type="text" value="1.1.1.1"/>		<input type="text"/>		
<input type="checkbox"/> Send Signature with order				
<input type="button" value="Preview Fax"/>		<input type="button" value="Send"/>		<input type="button" value="Cancel"/>
<input type="button" value="Clear Signature"/>				

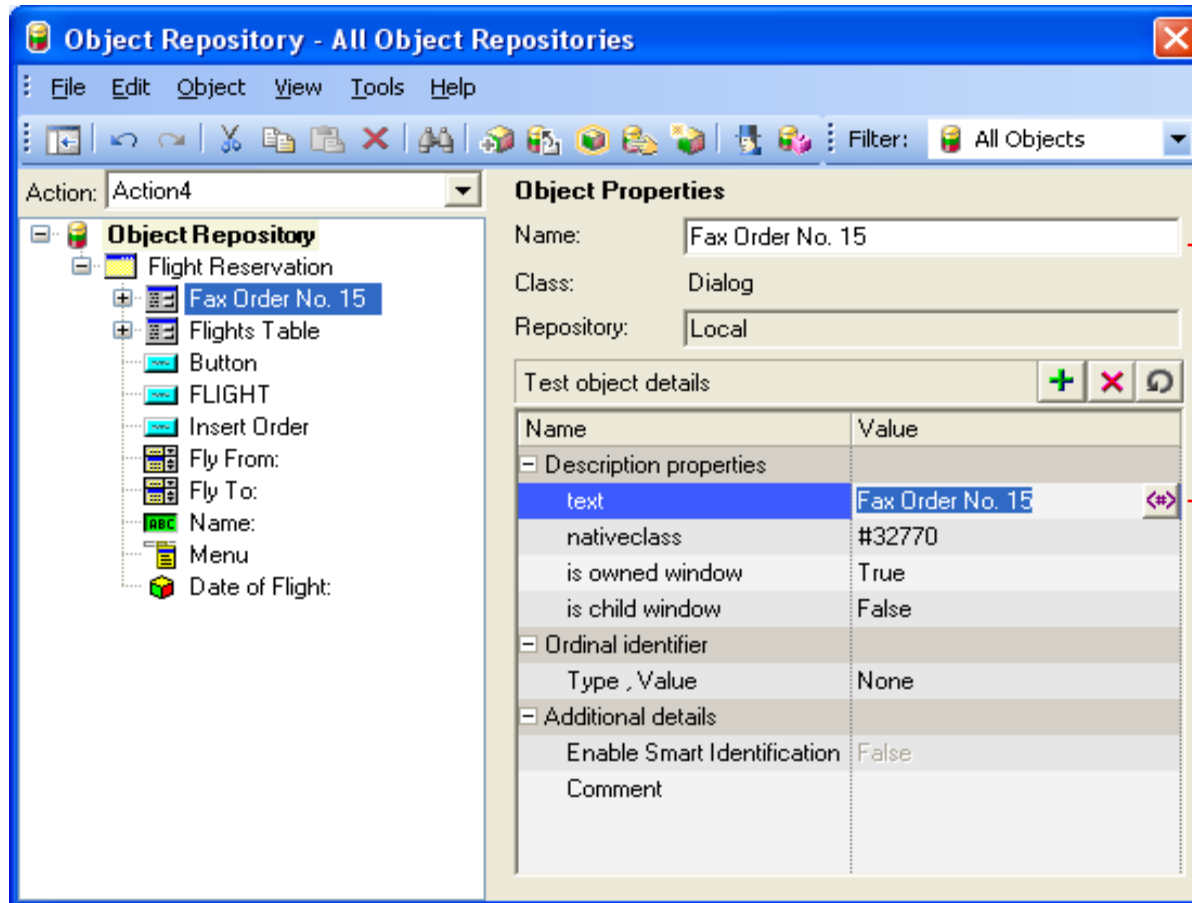
To add Regular Expression

■ Resources – Object Repository



Regular Expressions – Object Repository

- To handle this change the fax order window name in Object Repository



Name

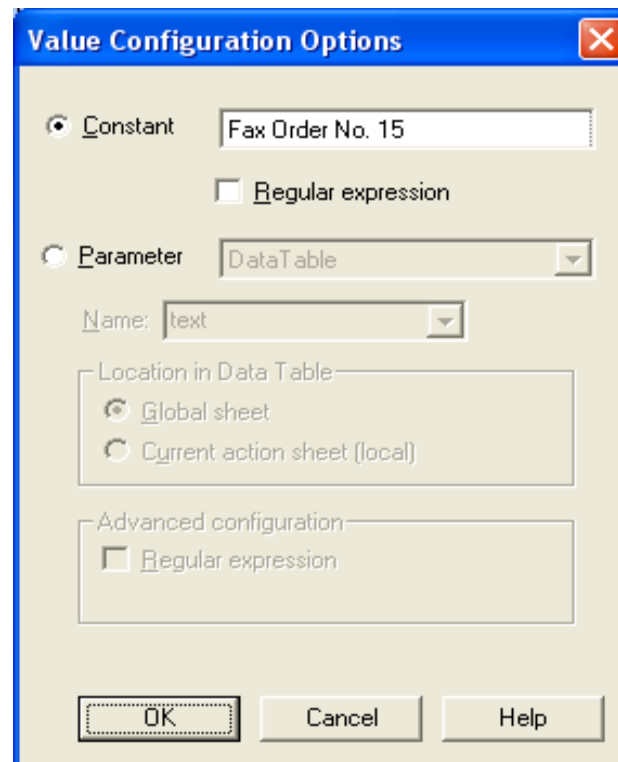
Configure Value button

Regular Expression – Object Repository

Select Description Properties – text

Click the Configure value button next to text

The following window will be displayed



The image shows a 'Value Configuration Options' dialog box. It has a blue title bar with a close button. The dialog is divided into two main sections: 'Constant' and 'Parameter'. The 'Constant' section is selected with a radio button, and it contains a text field with the value 'Fax Order No. 15' and an unchecked checkbox for 'Regular expression'. The 'Parameter' section is also visible, with a radio button, a dropdown menu showing 'DataTable', and a 'Name' dropdown menu showing 'text'. Below these are two expandable sections: 'Location in Data Table' with radio buttons for 'Global sheet' (selected) and 'Current action sheet (local)', and 'Advanced configuration' with an unchecked checkbox for 'Regular expression'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

Value Configuration Options

☒ **Constant**
☐ **Regular expression**

☐ **Parameter**
Name:

Location in Data Table
☒ **Global sheet**
☐ **Current action sheet (local)**

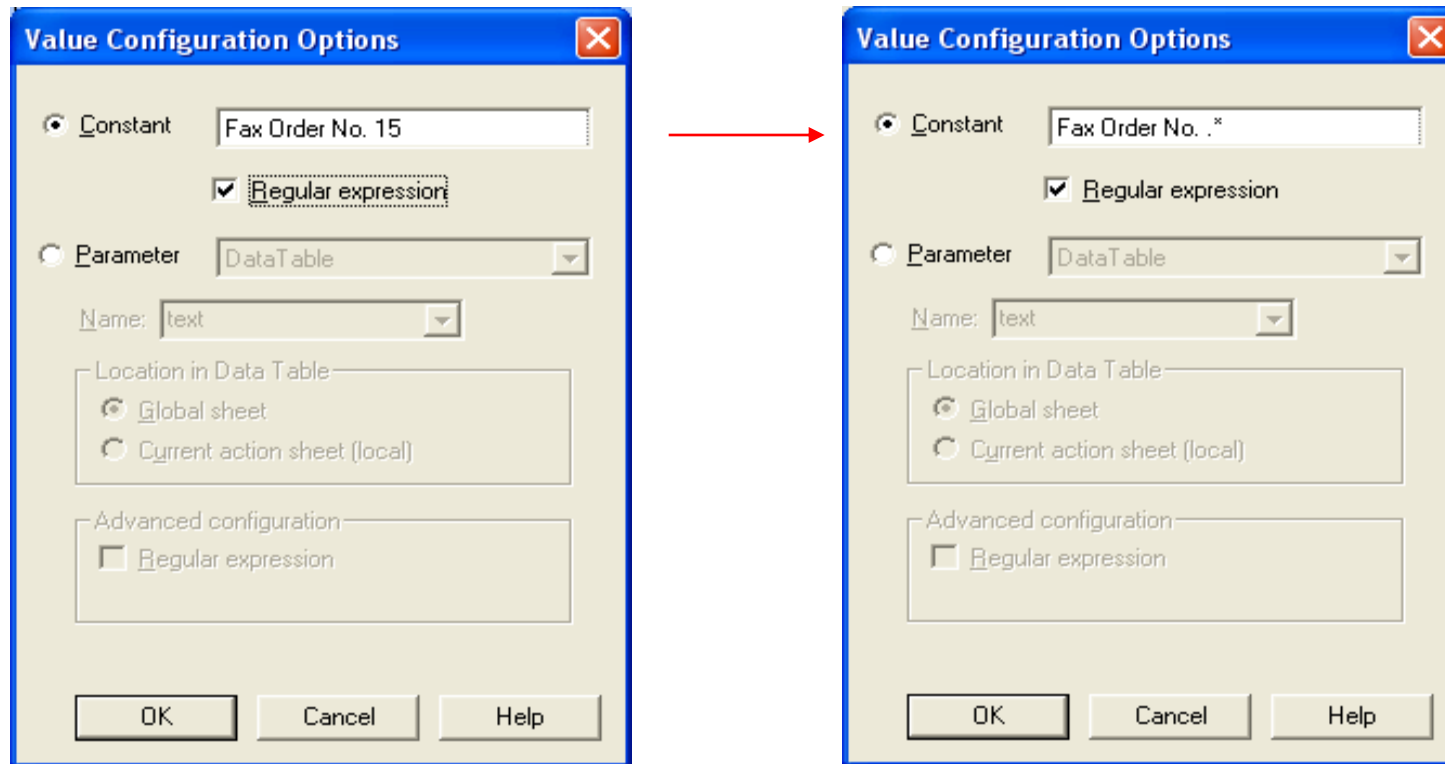
Advanced configuration
☐ **Regular expression**

OK **Cancel** **Help**

Regular Expression

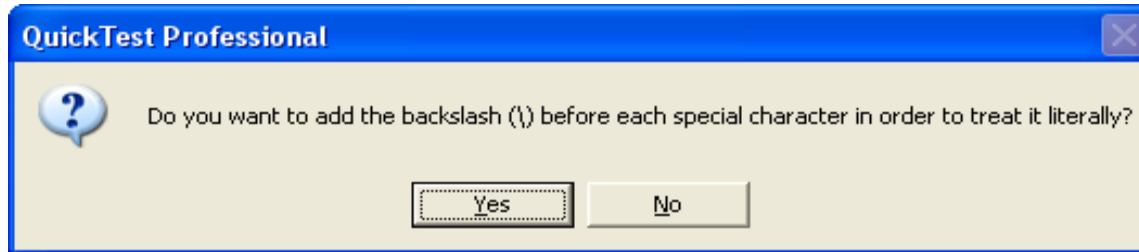
Change the name of the window as Fax Order No. .*

Click the Regular Expression Check box



Regular Expression

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



Regular Expression

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

changes

The screenshot shows the 'Object Repository - All Object Repositories' window. The left pane displays a tree structure under 'Object Repository' with 'Flight Reservation' expanded, showing 'Fax Order No. *' selected. The right pane shows 'Object Properties' for this object, with 'Name' set to 'Fax Order No. *', 'Class' set to 'Dialog', and 'Repository' set to 'Local'. Below this is a 'Test object details' table.

Name	Value
Description properties	
text	R.* Fax Order No. *
nativeclass	#32770
is owned window	True
is child window	False
Ordinal identifier	
Type , Value	None
Additional details	
Enable Smart Identification	False
Comment	

Changes

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)

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 DbClick step. Otherwise, the virtual object is ignored.

Steps for Creating Virtual Object

- In QuickTest, 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("7").Click  
Window("Calculator").WinButton("4").Click  
Window("Calculator").WinButton("2").Click  
Window("Calculator").WinButton("3").Click  
Window("Calculator").WinButton("+").Click  
Window("Calculator").WinButton("2").Click  
Window("Calculator").WinButton("4").Click  
Window("Calculator").WinButton("1").Click  
Window("Calculator").WinButton("=").Click
```

- The corresponding Keyword View will be as shown below.

Item	Operation	Value	Documentation
▼ Action1			
▼ Calculator			
7	Click		Click the "7" button.
4	Click		Click the "4" button.
2	Click		Click the "2" button.
3	Click		Click the "3" button.
+	Click		Click the "+" button.
2	Click		Click the "2" button.
4	Click		Click the "4" button.
1	Click		Click the "1" button.
=	Click		Click the "=" button.

Case Study of Virtual Object (contd.)

- Suppose button 7 is made a virtual object and we give the name 7. Now, the script will be displayed as shown below. The button “7” has been assigned as an virtual object.

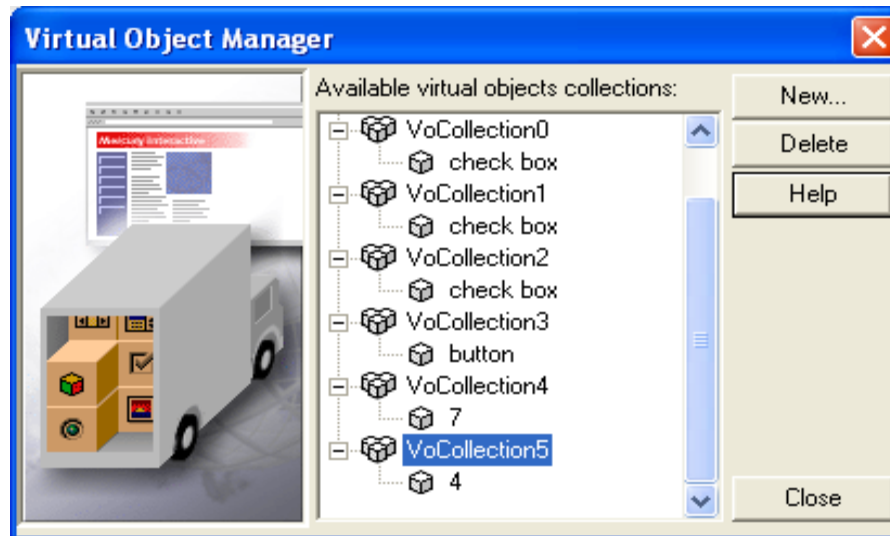
```
Window("Calculator_2").WinButton("7").VirtualButton("7").Click  
Window("Calculator_2").Move 454,309  
Window("Calculator_2").WinButton("5").Click  
Window("Calculator_2").WinButton("2").Click
```

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

▼ Calculator_2	Activate		Make the "Calculator_2" window active.
▼ 7			
7	Click		Click the "7" button.
▼ Calculator_2	Move	454,309	Move the "Calculator_2" window to the screen coordinates 454, 309.
5	Click		Click the "5" button.
2	Click		Click the "2" button.

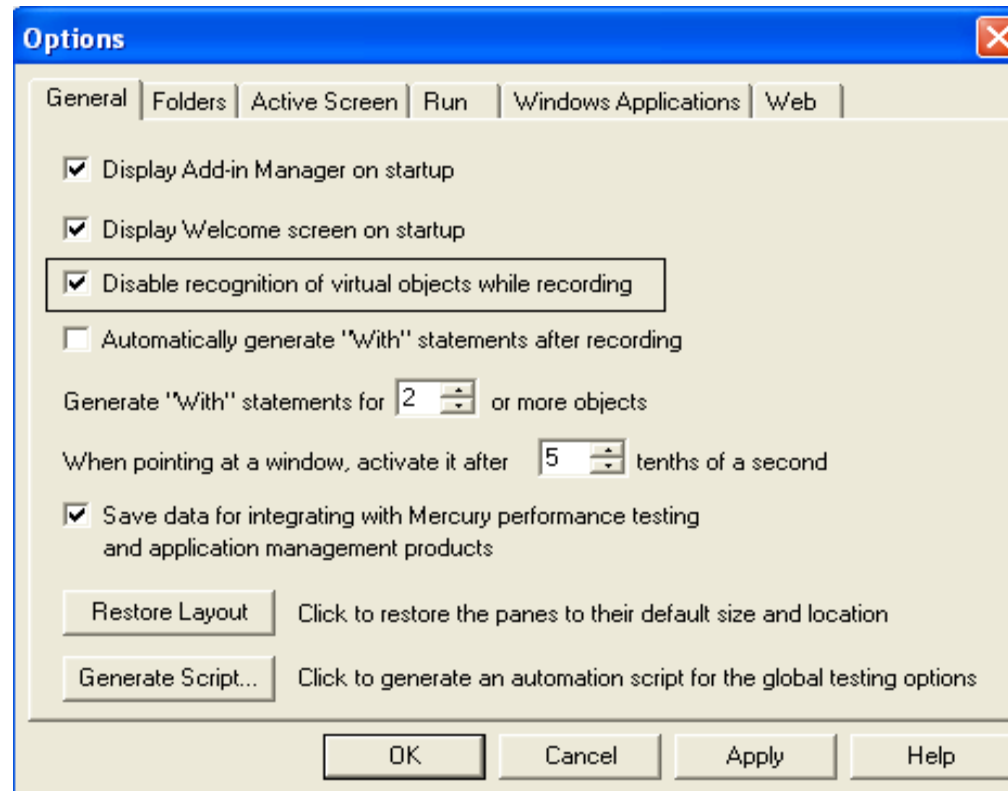
Removing Virtual Objects

- 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



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



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 DbClick 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.
- QuickTest does not support Virtual Objects for analog or low-level recording

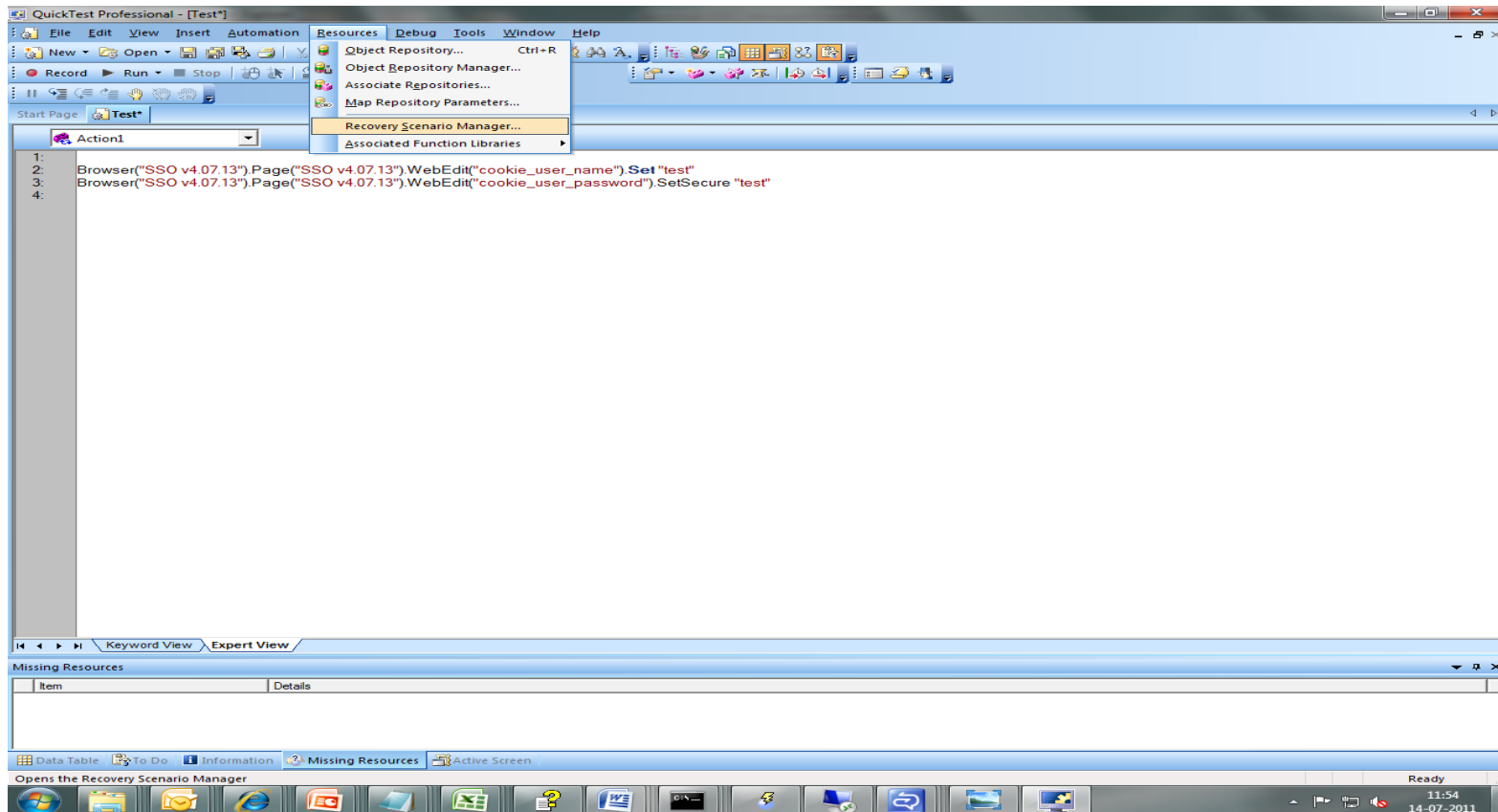
Recovery Scenarios

What is a Recovery Scenario?

- **Recovery Scenario is a mechanism by which QTP 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**

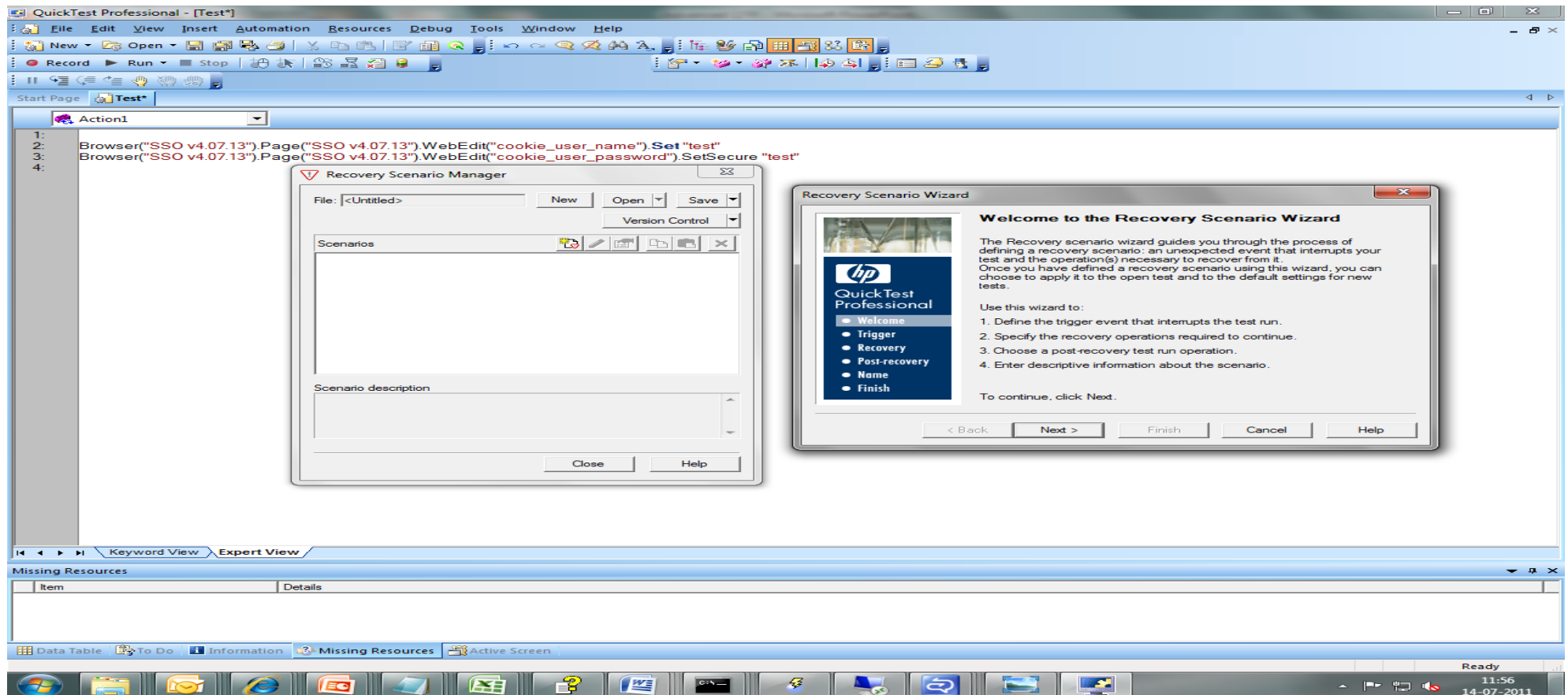
How to create a recovery scenario?

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



How to create a recovery scenario?

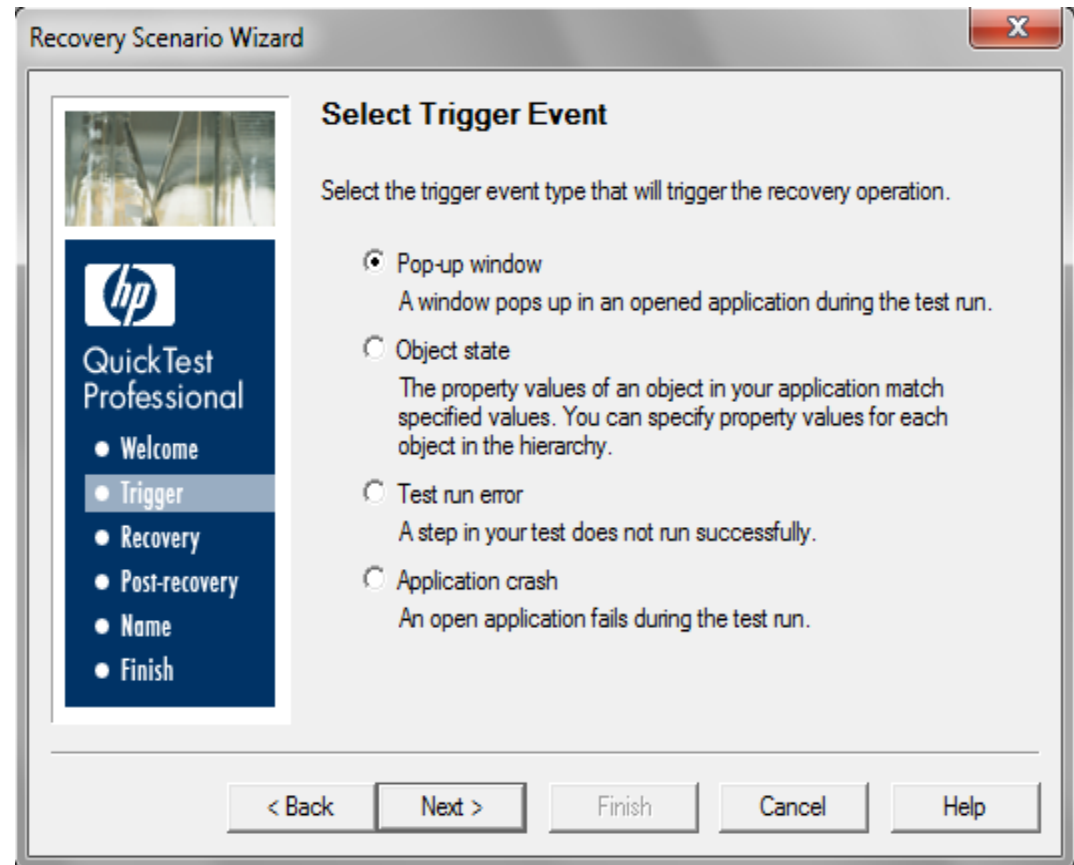
Recovery Scenario Wizards starts. Click Next on this screen.



How to create a Recovery Scenario?

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

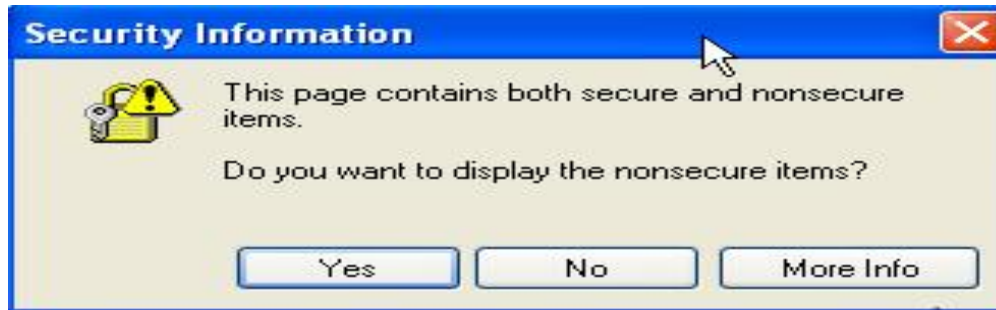
Recovery scenarios can be defined for unwanted Pop-up window, Object state, Test run error or application crash.



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



How to create a Recovery Scenario?

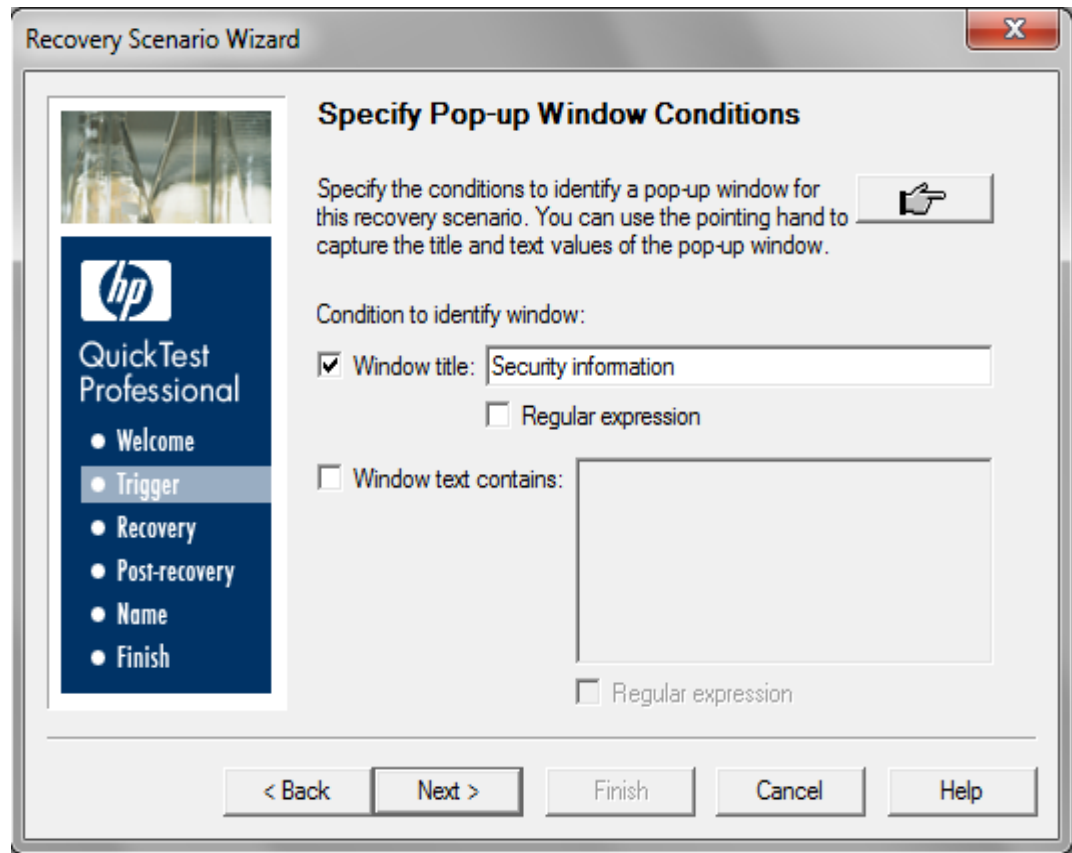
QTP captures the window title and window text.

Uncheck the checkbox "Window text contains".

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

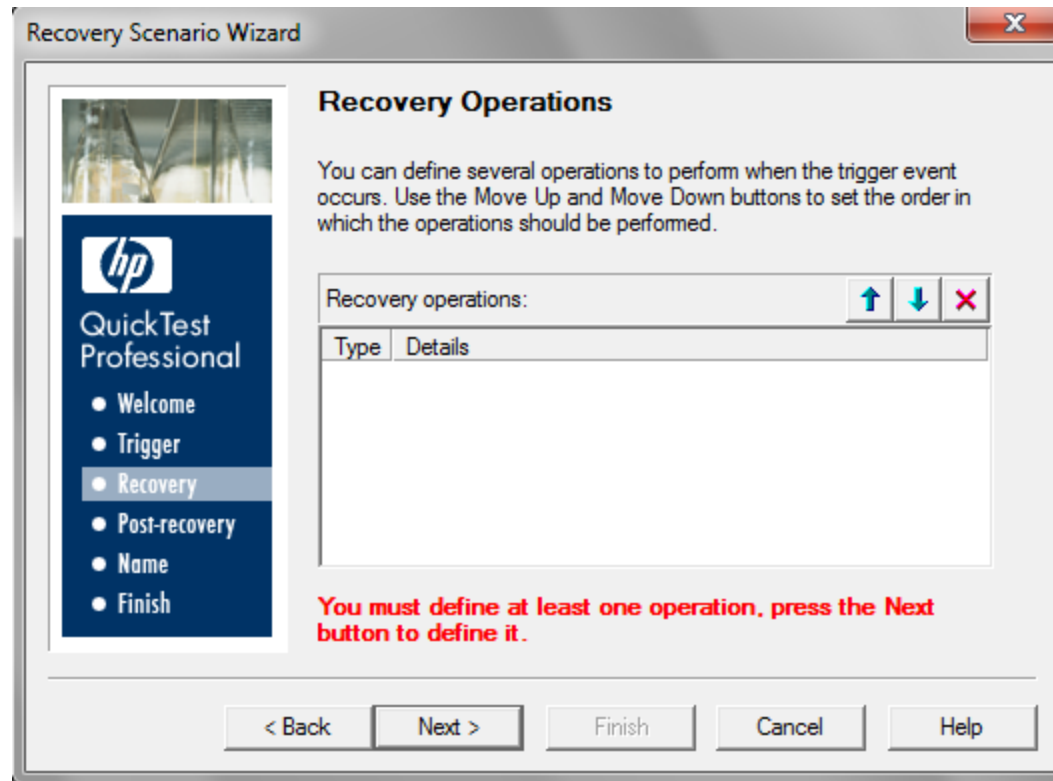
If the window title changes dynamically with some pattern, Click the checkbox "Regular Expression" and provide the pattern.

Click Next



How to create a Recovery Scenario?

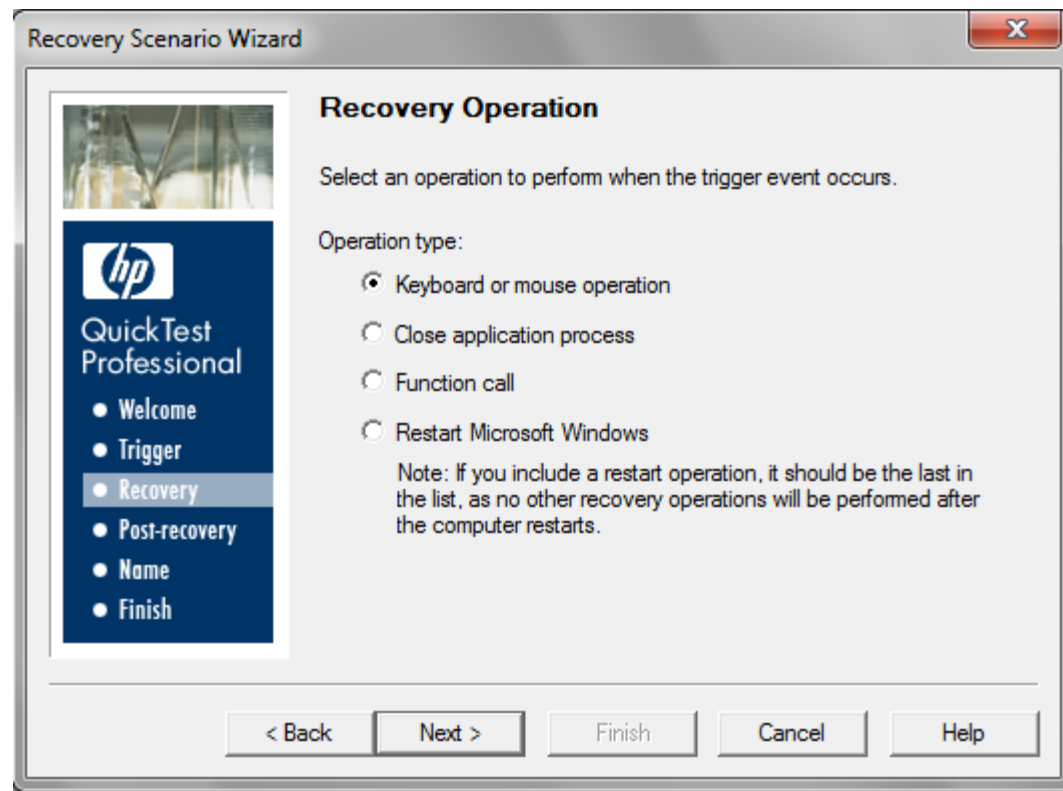
Next screen informs us that we should define the recovery operation to be done in order to handle this window.



How to create a Recovery Scenario?

Click Next button to view the Recovery Operation window.

Select the appropriate action to be performed.



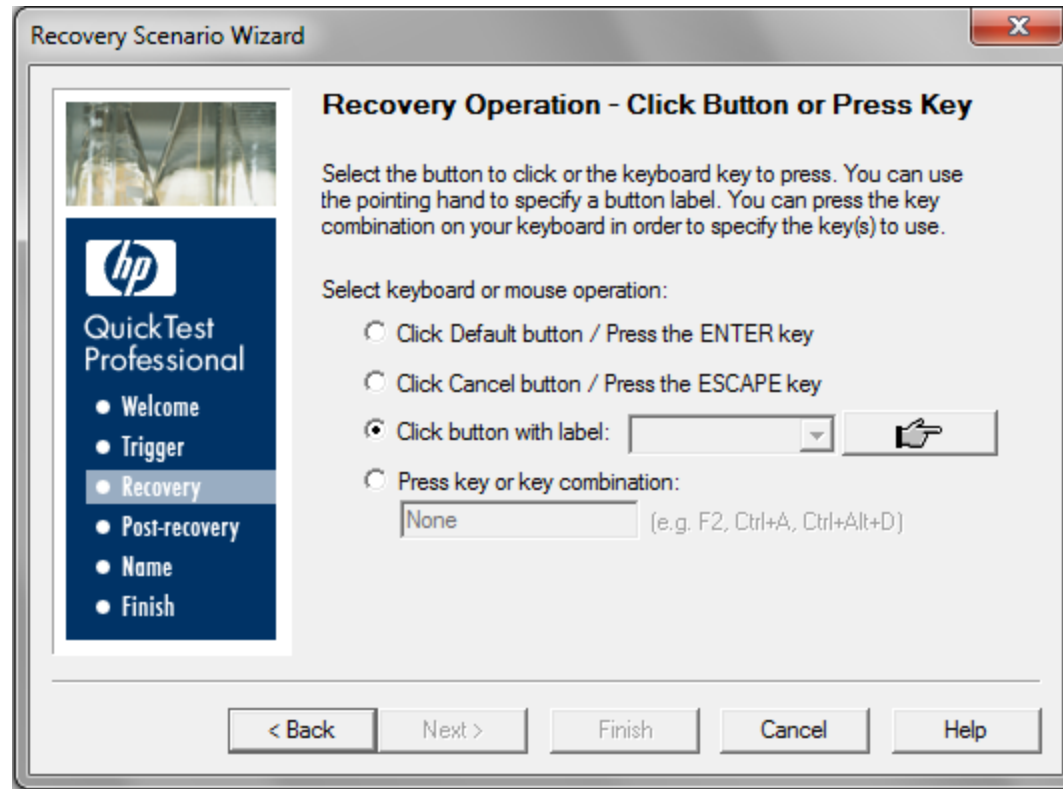
How to create a Recovery Scenario?

- **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 QTP to click using the hand icon in the following screen and click Next button.

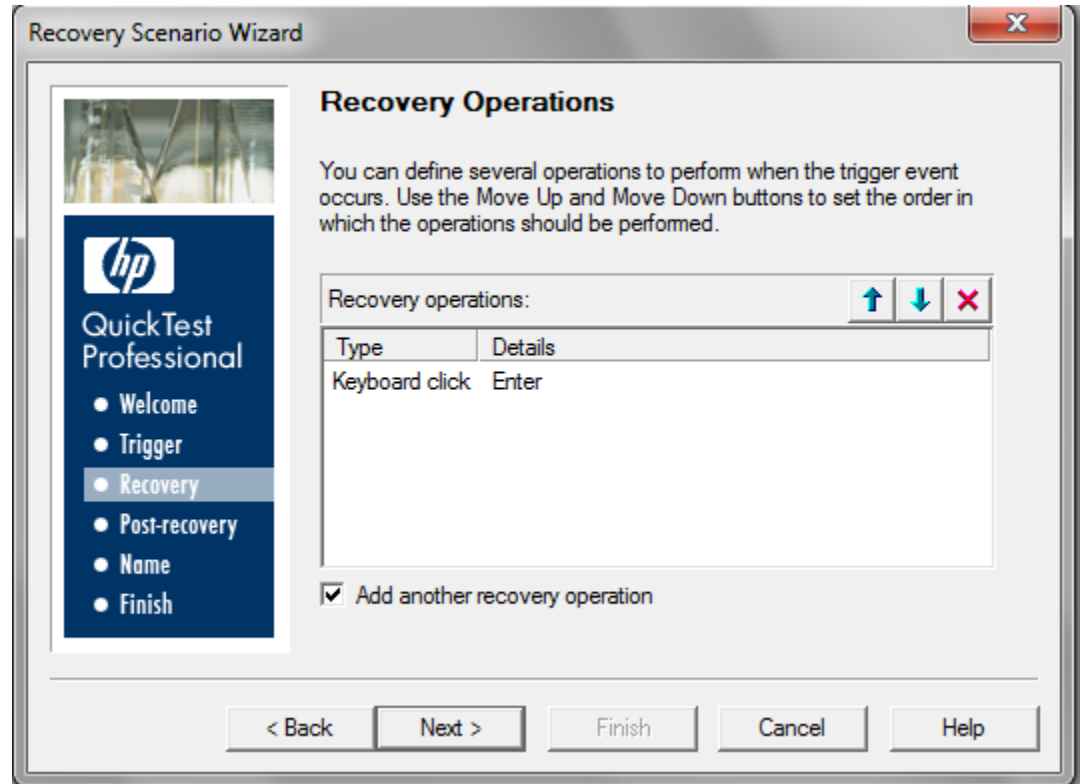


How to create a Recovery Scenario?

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"

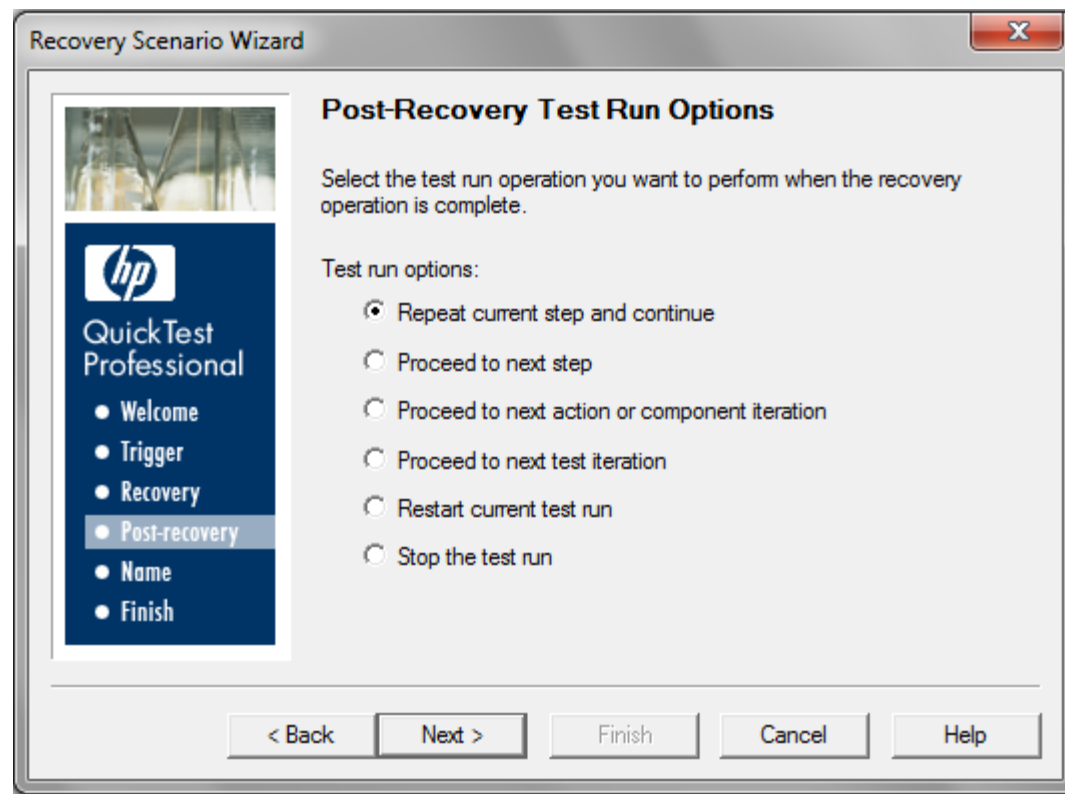


The screenshot shows the "Recovery Scenario Wizard" dialog box. On the left is a sidebar with the HP logo and a list of steps: Welcome, Trigger, Recovery (selected), Post-recovery, Name, and Finish. The main area is titled "Recovery Operations" and contains instructions: "You can define several operations 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." Below this is a table with the header "Recovery operations:" and two columns: "Type" and "Details". The table contains one row: "Keyboard click" in the Type column and "Enter" in the Details column. To the right of the table are three buttons: a blue up arrow, a blue down arrow, and a red X. At the bottom of the main area is a checkbox labeled "Add another recovery operation" which is checked. At the very bottom of the dialog are five buttons: "< Back", "Next >", "Finish", "Cancel", and "Help".

Type	Details
Keyboard click	Enter

How to create a Recovery Scenario?

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



How to create a Recovery Scenario?

Since the recovery scenario is kicked off only when the error is about to be thrown since QTP 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"

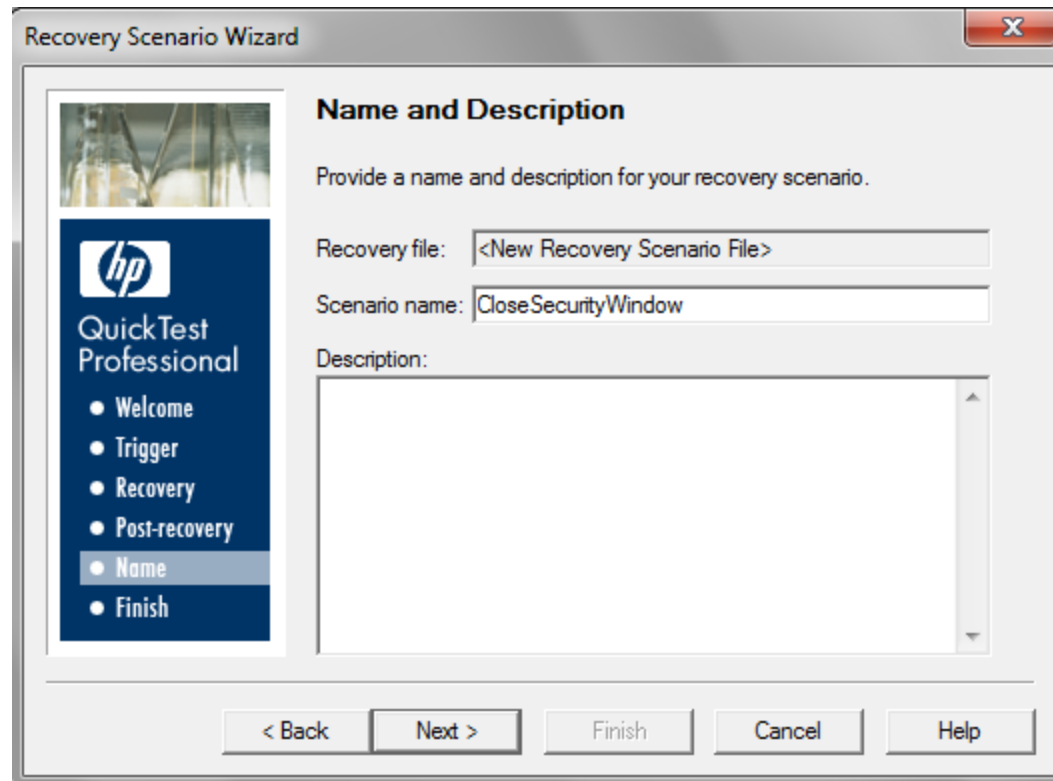
How to create a Recovery Scenario?

- 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".



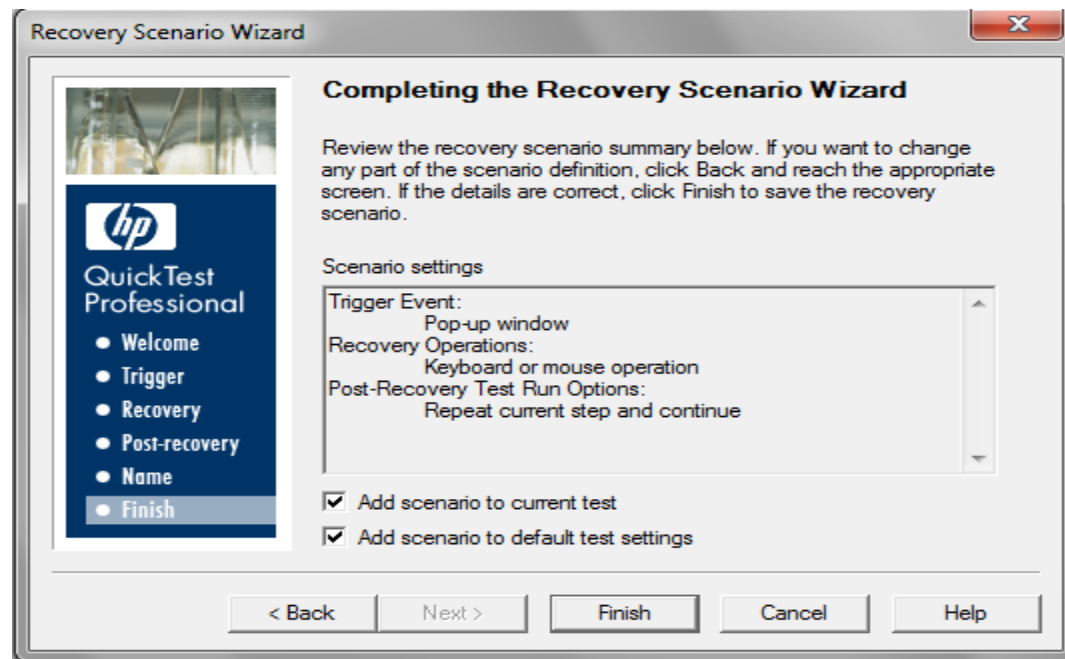
The screenshot shows the 'Recovery Scenario Wizard' dialog box. On the left is a sidebar with the HP logo and a list of steps: Welcome, Trigger, Recovery, Post-recovery, Name (selected), and Finish. The main area is titled 'Name and Description' and contains the instruction 'Provide a name and description for your recovery scenario.' Below this are three input fields: 'Recovery file:' with a dropdown menu showing '<New Recovery Scenario File>', 'Scenario name:' with a text box containing 'CloseSecurityWindow', and 'Description:' with a large empty text area. At the bottom are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

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 QTP settings as well.

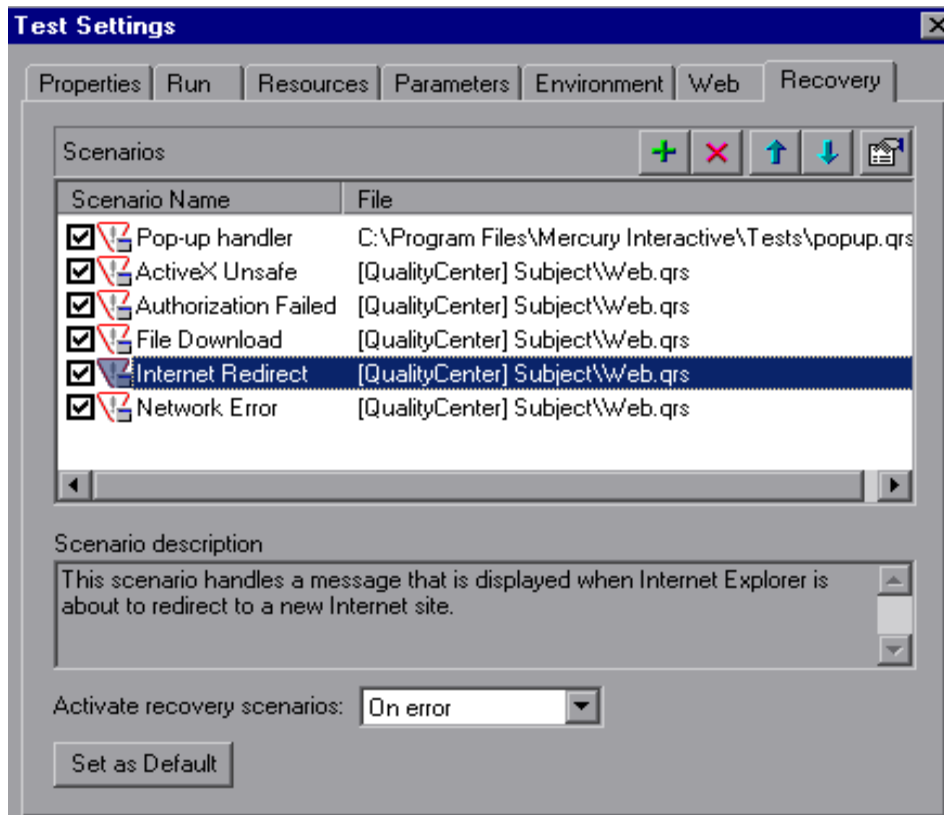
Click "Finish" and follow the further steps to save the recovery scenario to a recovery file








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.



Button	Description
	Opens the Add Recovery Scenario dialog box, which enables you to associate one or more recovery scenarios with the component. For more information, see Defining and Using Recovery Scenarios .
	Removes the selected recovery scenario from the component.
	Moves the selected scenario up in the list, giving it a higher priority during the component run session.
	Moves the selected scenario down in the list, giving it a lower priority during the component run session.
	Displays summary properties for the selected recovery scenario in read-only format. For more information, see Defining and Using Recovery Scenarios .

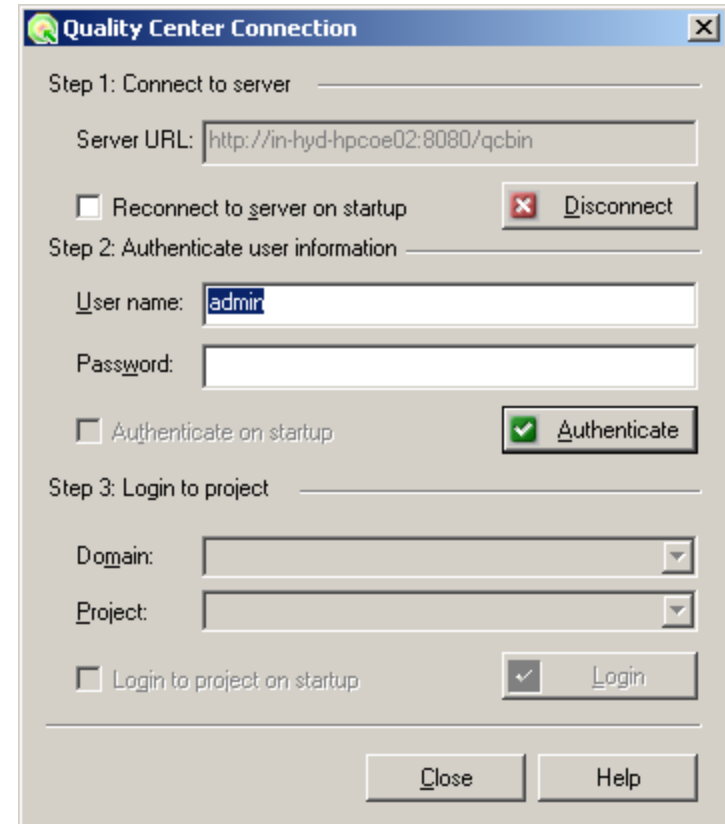
QC Integration

Prerequisites to connect QTP with QC

- Check Allow other Mercury products to run tests and components present under Tools > Options > Run in QTP
- If you are running the tests on the same computer where you have QC client installed, then you will need:
 - QTP Connectivity Add-In
 - QTP Add-in
- If you are running the tests on the different computer than where you have QC client installed, then you will need:
 - QTP Add-in where QC client is installed.
 - QTP Add-in and QC connectivity Add-in where QTP is installed.
- QC connectivity can be found at QC server URL > 'Add-Ins Page' link > 'QC Connectivity' link > 'Download Add-in'
- QTP Add-in can be found at QC server URL > 'Add-Ins Page' link > 'More QC Add-ins' link > Download and install QTP Add-in according to its version

Connecting Quick Test to Quality Center

- Start QTP and from File option select Quality Center Connection.
- 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



The image shows a 'Quality Center Connection' dialog box with three steps:

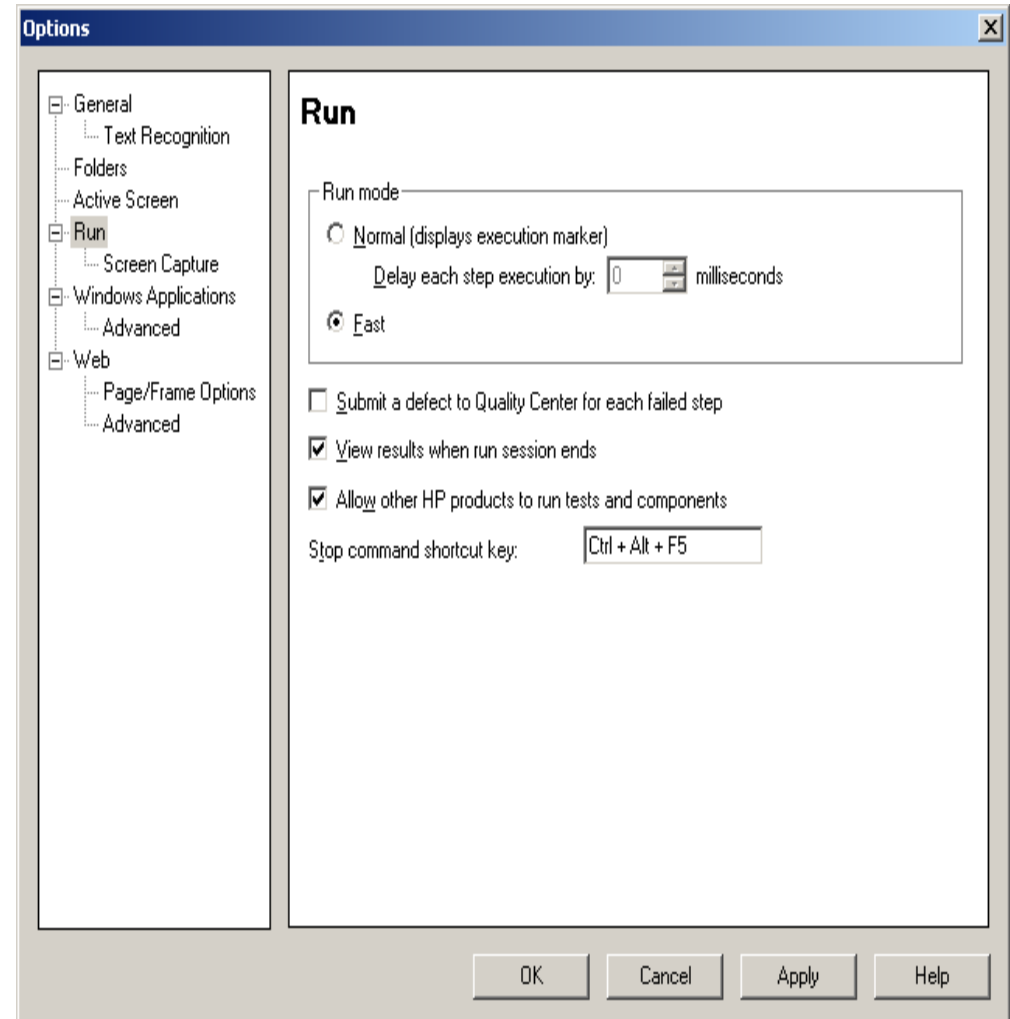
- Step 1: Connect to server**
 - Server URL: `http://in-hyd-hpcoe02:8080/qcbn`
 - ☐ Reconnect to server on startup
 - ☒ Disconnect
- Step 2: Authenticate user information**
 - User name: `admin`
 - Password: (empty)
 - ☐ Authenticate on startup
 - ☒ Authenticate
- Step 3: Login to project**
 - Domain: (empty)
 - Project: (empty)
 - ☐ Login to project on startup
 - ☒ Login

Buttons at the bottom: Close, Help

Enabling Quality Center to Run Tests on a Quick Test Computer

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

- Open Quick Test.
- Choose Tools > Options or click the Options toolbar button. The Options dialog box opens.
- Click the Run tab.
- Select the “Allow other Mercury products to run tests and components” check box.



Running a Test Stored in a Quality Center Project

- Test can be executed either from QC Test lab or from QTP
- To execute the Test from QTP
 - 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.

Run

Results Location | Input Parameters

Write run results to:

☒ New run results in Quality Center project

Project name: QTP_Quality_Test

Run name: Run 8-2 11-14-28

Test set: default Instance: 1

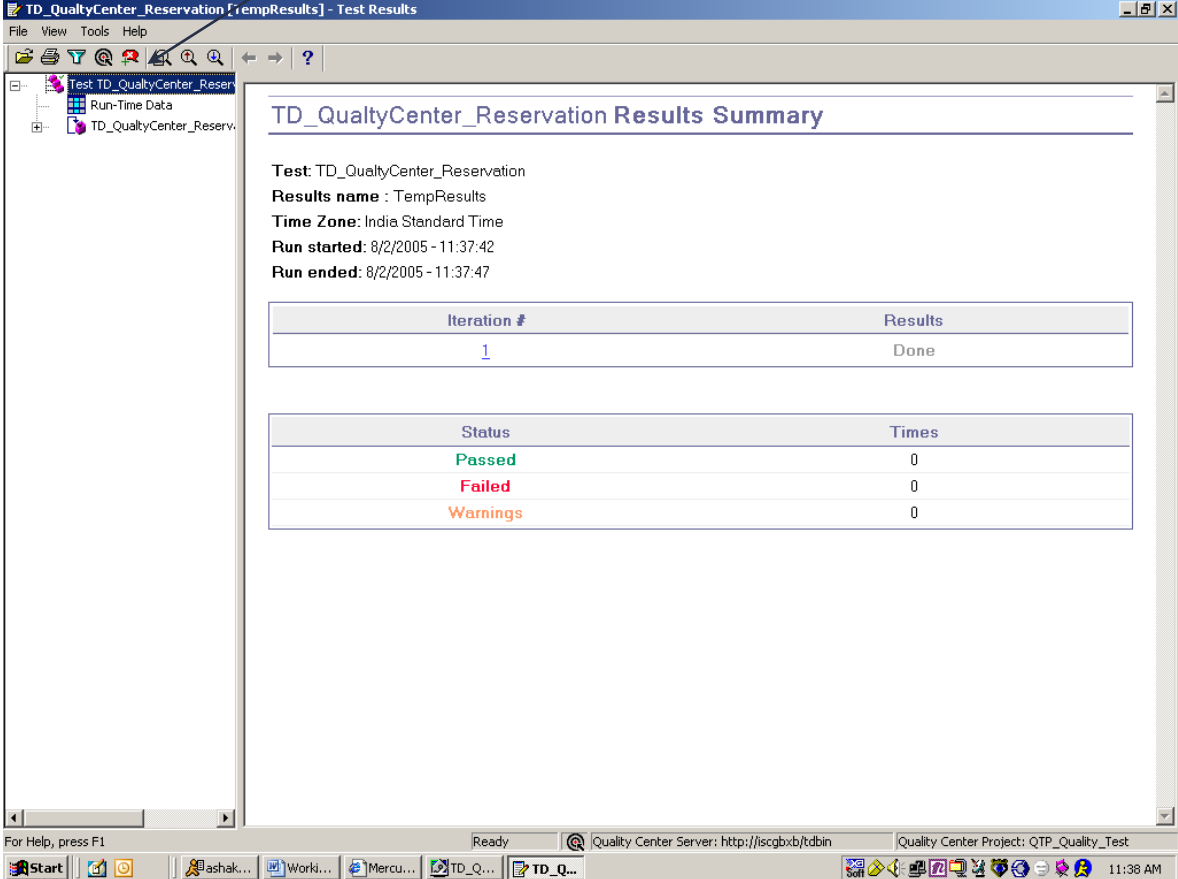
☐ Temporary run results folder (not saved in the project)

OK Cancel Help

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_QualityCenter_Reserv
Run-Time Data
TD_QualityCenter_Reserv.

TD_QualityCenter_Reservation Results Summary

Test: TD_QualityCenter_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

For Help, press F1

Ready

Quality Center Server: http://iscgbxb/tdbin

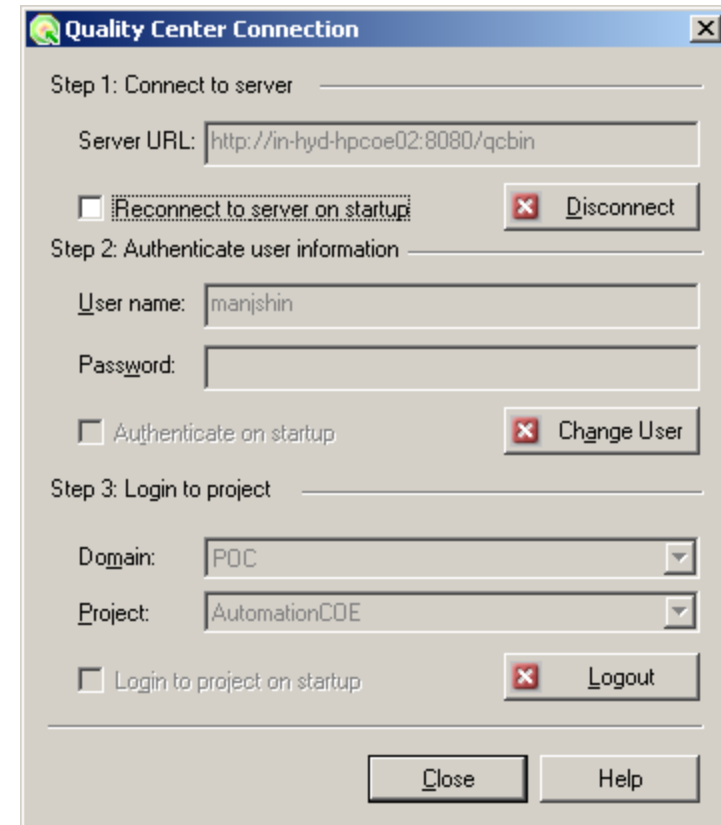
Quality Center Project: QTP_Quality_Test

Start | Bashak... | Worki... | Mercu... | TD_Q... | TD_Q...

11:38 AM

Disconnecting Quick Test to Quality Center

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



The image shows a 'Quality Center Connection' dialog box with three steps:

- Step 1: Connect to server**
 - Server URL:
 - ☐ Reconnect to server on startup
 -
- Step 2: Authenticate user information**
 - User name:
 - Password:
 - ☐ Authenticate on startup
 -
- Step 3: Login to project**
 - Domain:
 - Project:
 - ☐ Login to project on startup
 -

At the bottom are and .

People matter, results count.

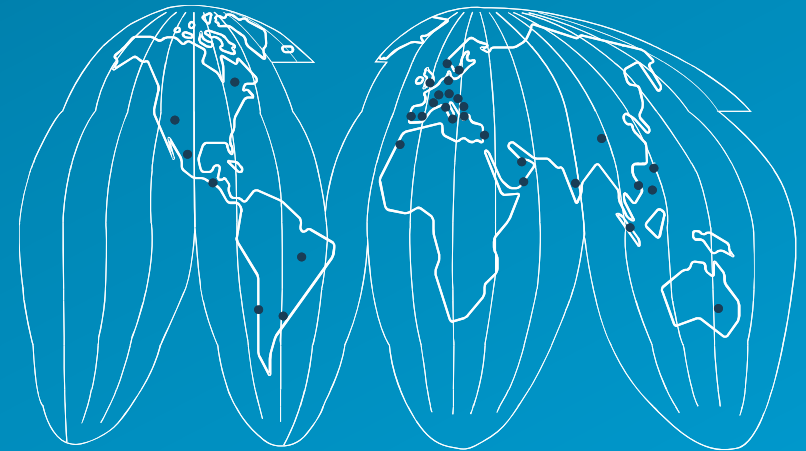


About Capgemini

With more than 120,000 people in 40 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2011 global revenues of EUR 9.7 billion.

Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Rightshore® is a trademark belonging to Capgemini



www.capgemini.com

