Synchronization and Checkpoints in UFT

# Synchronization

Synchronization and waits are features in Unified Functional Testing (UFT) that help ensure that a test script and the application under test (AUT) are in sync during a test run.

Synchronization issues can occur when there is a timing mismatch between the test script and the AUT. Some common causes of synchronization issues include: Slow loading of web elements, Dynamic content changes, Client-server communication delays, Variations in application performance, and Resource-intensive operations.

## Solving synchronization issues in UFT

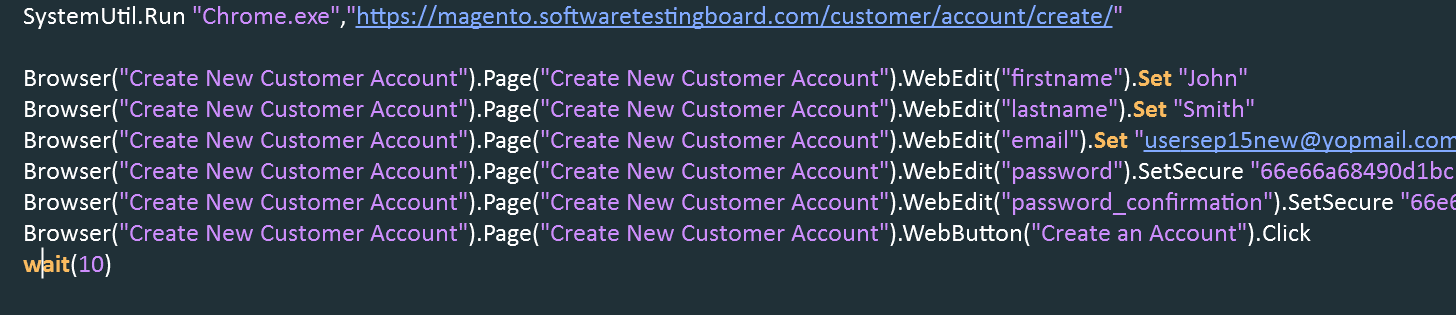
1. Wait Statement
2. Set Synchronization timeout
3. Wait Property
4. Using Sync method
5. Using Exist method
6. Set Browser navigation timeout

### Wait Statement

The **Wait**method can be used to wait for the specified time. The Wait method accepts time in second as input.

For example, when you open the browser and navigate to an URL, it may take 3 to 10 seconds to load the login page completely. You can not enter login credentials unless the page is loaded completely.  
So you can add a wait statement that will wait for 10 seconds.

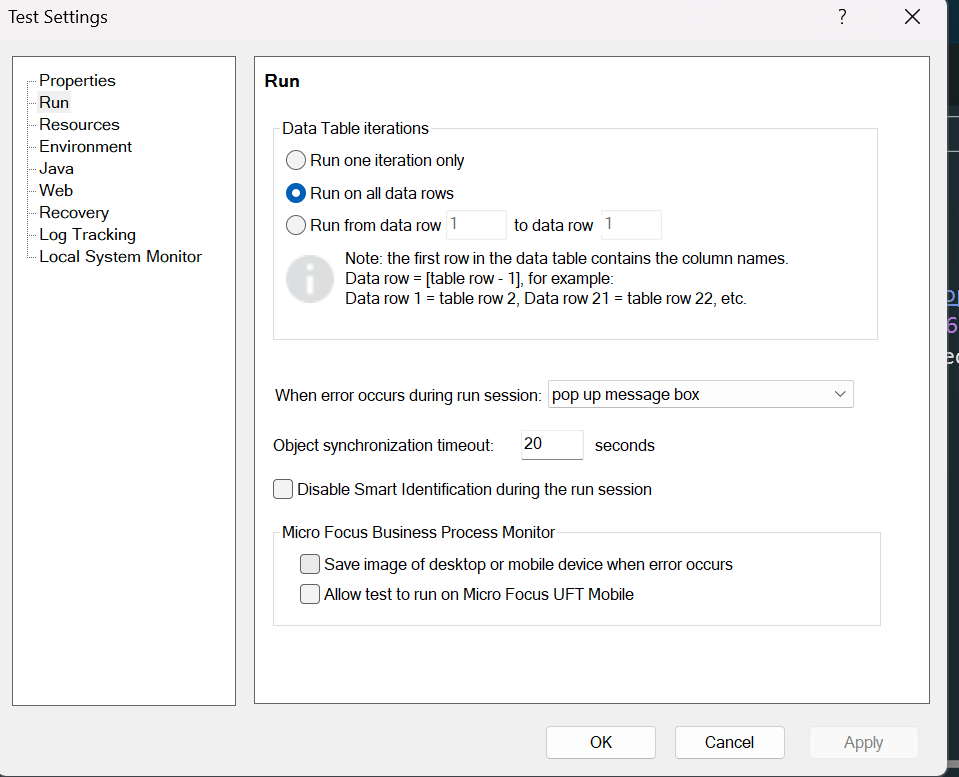
Since you are adding a hard wait of 10 seconds. It means that even if the page gets loaded within 4 seconds, UFT will still continue to wait for 6 more seconds. Hence it is not a good approach to insert a hard wait where the wait time varies between 3 to 10 seconds. The hard wait should not be used where the wait time exceeds more than three seconds.



### Set syncronization timeout

Set the global timeout to find all the objects using object synchronization timeout.

File 🡪 Settings 🡪 Run





Default value is 20 seconds

## Wait Property

We should always use dynamic wait rather than hard wait where we are not sure how much time an object would take to load. In order to use dynamic wait in UFT, we can use **WaitProperty** method.

**Syntax**

object.**WaitProperty**(PropertyName, PropertyValue, [TimeOut])

|  |  |
| --- | --- |
| Parameter | Description |
| PropertyName | Required. A String value. The name of the property whose value is checked |
| PropertyValue | Required. A Variant. The value to be achieved before continuing to the next step |
| TimeOut | TimeOut Optional. A long integer value. Time in milliseconds. After which UFT continues to the next step if the specified value is not achieved. If no value is specified, UFT uses the time set in the Object Synchronization Timeout option in the Run pane of the Test Settings dialog box. |

Browser("Create New Customer Account").Page("Create New Customer Account").WebEdit("firstname").WaitProperty("name", "First Name\*",10000)

### Using Sync Method

The Sync method is designed for web environments, ensuring the page has completed loading all its elements before proceeding with the next steps.

This will wait for the page to load completely, ensuring all elements are available for interaction.

Browser("Create New Customer Account").Page("Create New Customer Account").Sync

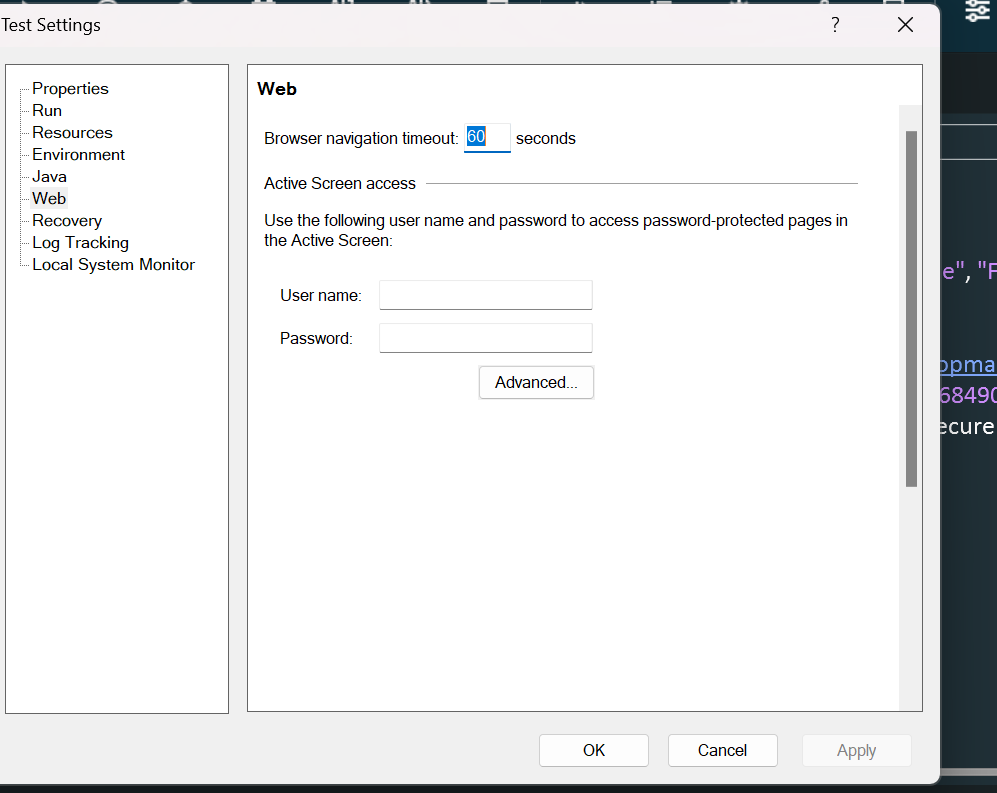
### Using Exist Method

Exist method will check for the existence of the object. It returns true if the object exists and false if the object is not found.

**If** Browser("Create New Customer Account").Page("Create New Customer Account").WebEdit("firstname").Exist **Then**

### Set Browser navigation timeout

**For Web objects only:** Increase the **Browser Navigation Timeout**(**File > Settings > Web** node).





# CheckPoints in UFT

UFT One enables you to add checks to your test or component. A **checkpoint** is a verification that compares the current value for specified properties or current state of other characteristics of an object with the expected value or characteristics. This helps you to identify whether your application is functioning correctly. For example, you can perform standard checkpoints to check that the actual object property values conform to the expected values, and you can perform bitmap checkpoints to check that the visible parts of your application are displayed correctly.

## Different types of Checkpoints

* Standard Checkpoint
* Bitmap Checkpoint
* Text Checkpoint
* Text Area Checkpoint
* Accessibility Checkpoint
* Page Checkpoint
* Database Checkpoint
* XML Checkpoint

### Standard Checkpoint

Checks property values of an object in your application. For example, you can check that a radio button is activated after it is selected or you can check the value of an edit box.

Standard checkpoints compare the expected values of object properties to the object's current values during a run session.

### Bitmap Checkpoint

Checks the value of an image in your application. For example, you can check that a selected image's source file is correct.

You create an image checkpoint by inserting a standard checkpoint on an image object.

Image checkpoints are supported for the Web add-in environment.

UFT One enables you to check that the visible parts of your application are displayed correctly by comparing bitmaps of objects in your application to bitmaps captured previously and stored with the test or component.

Bitmap checkpoints enable you to do the following:

* **Compare an entire object or areas within an object.** For example, suppose you have a Web site that can display a map of a city that the user specifies. The map has control keys for zooming. You can zoom in on a map, and then insert a bitmap checkpoint on the zoomed-in map to check that the map zooms in correctly.
* **Locate a specified image within an object.** For example, suppose you want to check that your company logo is displayed on your Web page.

#### Fine tuning bitmap checkpoint

When running a bitmap checkpoint, UFT One compares the area that you are checking in the application with the bitmap stored in the checkpoint, pixel by pixel. By default, if any pixels are different, the checkpoint fails.

1. **RGB Tolerance- The RGB (Red, Green, Blue) tolerance determines the percent by which the RGB values of the pixels in the runtime bitmap can differ from those of the expected bitmap and allow the checkpoint to pass.**

For example, a bitmap checkpoint on identical bitmaps could fail if different display drivers are used when you create your checkpoint and when you run your test. Suppose one display driver displays the color white as RGB (255, 255, 255) and another driver displays the color white as RGB (231, 231, 231). The difference between these two values is about 9.4%. By setting the **RGB tolerance** to 10%, your checkpoint will pass when running your test with either of these drivers.

1. **Pixel Tolerance - The pixel tolerance determines the number or percentage of pixels in the runtime bitmap that can differ from those in the expected bitmap and allow the checkpoint to pass.**

For example, suppose the expected bitmap has 4000 pixels. If you define the pixel tolerance to be 50 and select the **Pixels** radio button, up to 50 pixels in the runtime bitmap can be different from those in the expected bitmap and the checkpoint passes.

If you define the pixel tolerance to be 5 and select the **Percent**radio button, up to 200 pixels (5 percent of 4000) in the runtime bitmap can be different from those in the expected bitmap and the checkpoint passes.

### Text Checkpoint

Checks that a text string is displayed in the appropriate place in an application. For example, suppose a Web page displays the sentence **Flight departing from New York to San Francisco**. You can create a text checkpoint that checks that the words "New York" are displayed between "Flight departing from" and "to San Francisco".

Example:

Click on Account Information and verify Edit Account Information text

### TextArea Checkpoint

Enables you to check that a text string appears within a defined area in a Windows application, according to specified criteria.

When checking text displayed in a Windows-based application, it is often advisable to define a text area larger than the actual text you want UFT One to check.

Summarizes the selected text for the checkpoint. It displays the text you selected when creating the checkpoint, plus the text before and after it. UFT One automatically displays the checked text in red, and the text before and after the checked text in blue.

### Accessibility checkpoint

In UFT, accessibility checkpoints help identify areas of a website that may not meet the World Wide Web Consortium (W3C) Web Content Accessibility Guidelines. You can add accessibility checkpoints to pages or frames

Example:

* Alternative text is given for images
* Verify links are not broken

### Page Checkpoint

You can use page checkpoints to check statistical information about your Web pages. These checkpoints check the links and the sources of the images on a Web page. You can also instruct page checkpoints to include a check for broken links.

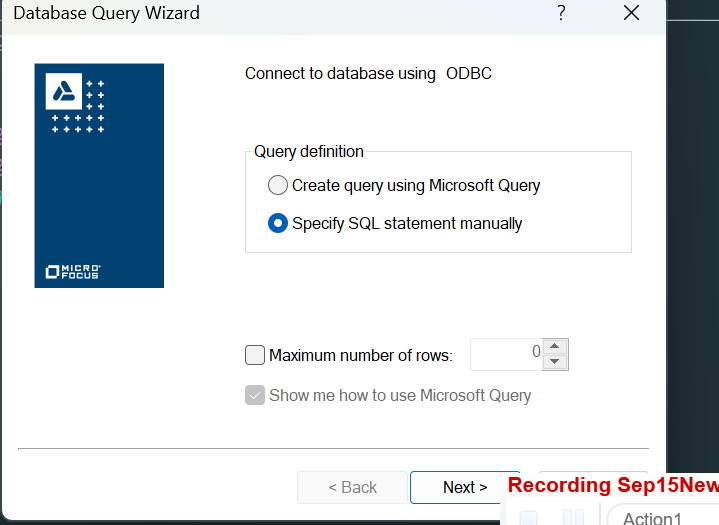
**Automatic Page Checkpoints -** You can instruct UFT One to create automatic page checkpoints for every page during a recording session by selecting the Create a checkpoint for each Web page while recording check box in the Web > Advanced pane of the Options dialog box (Tools > Options > GUI Testing tab > Web > Advanced node). By default, the automatic page checkpoint includes the checks that you select from among the available options in the Web > Advanced pane.

### Database Checkpoint

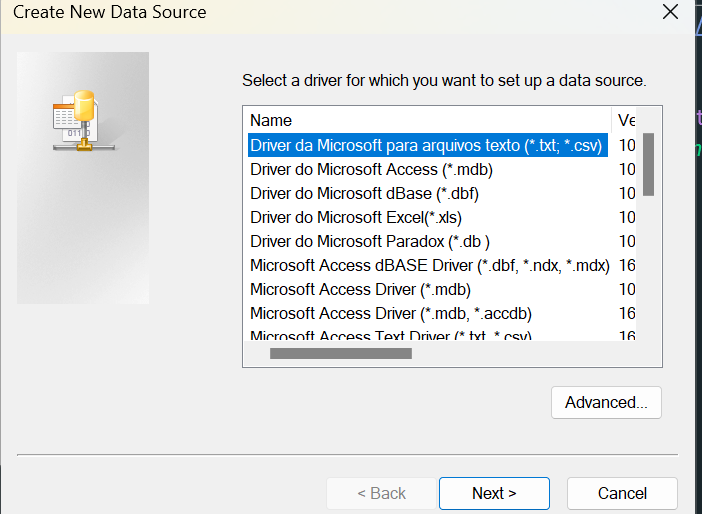
You can use database checkpoints to check databases accessed by your application, and to detect defects. To do this, you define a query on your database. Then you create a database checkpoint that checks the results of the query.

You create a database checkpoint based on the results of the query (**result set**) you defined on a database. You can create a check on a database to check the contents of the entire result set, or a part of it. UFT One captures the current data from the database, saves this information as **expected data**, and inserts a database checkpoint step.

1. Select add Database checkpoint



1. Click on Create for DSN



1. Select Excel driver
2. Select a location to store DSN file
3. Select workbook that will be the source

Provide the SQL query

Sample: Select \* from [Sheet1$]

### XML Checkpoint

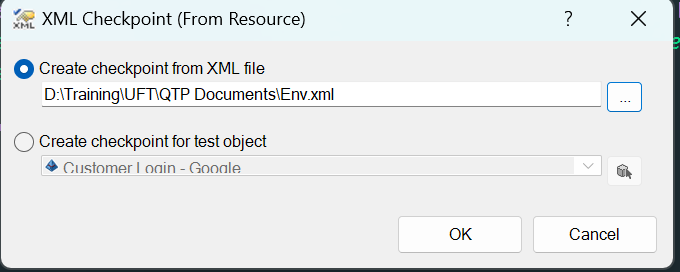
An XML checkpoint is a verification point that compares a current value for a specified XML element, attribute and/or value with its expected value. During a run session, UFT One compares the expected results of the checkpoint to the current results.

You can create the following types of XML checkpoints:

* **XML Web Page/Frame Checkpoint.** Checks an XML document within a Web page or frame.
* **XML File Checkpoint.** Checks a specified XML file.

To create XML Checkpoint

1. Design 🡪 Checkpoint 🡪 XML checkpoint
2. Select the xml file to validate



1. Select parameters to test
2. Add the checkpoint

# Output Values in UFT

Your test or component can retrieve values and store them in output value objects. You can then use these values as input at a later stage in a run session.

An **output value** step is a step in which one or more values are captured at a specific point in your test or component and stored for the duration of the run session. The values can later be used as input at a different point in the run session.

You can output the property values of any object. You can also output values from text strings, table cells, databases, and **.xml** documents.

When you create output value steps, you can determine where the values are stored during the run session and how they can be used. During the run session, UFT One retrieves each value at the specified point and stores it in the specified location. When the value is needed later in the run session, UFT One retrieves it from this location and uses it as required.

Output values are stored only for the duration of the run session. When the run session is repeated, the output values are reset.

## Types of output values

* Standard Output Values
* File Content Output Values
* Table Output Values
* Text and Text Area Output Values
* Database Output Values
* XML Output Values

When you define an output value, you can specify where and how each value is stored during the run session.

* Run-time Data Table
* Environment Variables

### Standard output value

1. Login to flight
2. Place an order by capturing passenger name
3. Create another action to search order
4. Use the data table data

WpfWindow("Micro Focus MyFlight Sample").WpfButton("NEW SEARCH").Click

WpfWindow("Micro Focus MyFlight Sample").WpfTabStrip("WpfTabStrip").Select "SEARCH ORDER"

WpfWindow("Micro Focus MyFlight Sample").WpfEdit("byNameWatermark").Set DataTable("passengerName", dtGlobalSheet)

WpfWindow("Micro Focus MyFlight Sample").WpfButton("SEARCH").Click

passName = WpfWindow("Micro Focus MyFlight Sample").WpfTable("ordersDataGrid").GetCellData(1,0)

msgbox passName

### File Content Output Value

You can use file content output values to output the contents from any of the following file types:

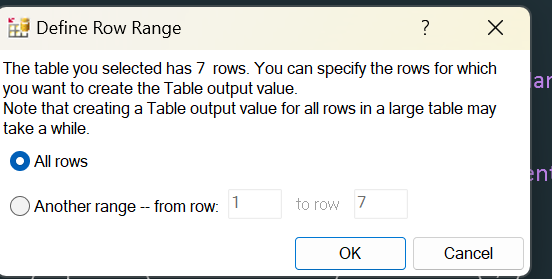
* HTML files
* Word (.doc) files
* Text files
* PDF files
* RTF files

You can create output values from the entire contents of a file, or from a part of it. During the run session, UFT One retrieves the current data from the file and outputs the values according to the settings that you specified.

1. Create flight booking and save the order number in a file
2. In Search flight test, add File content output value to fetch from file to datatable
3. Enter the order number in the search box

### Table Output values

Table output values are a subset of standard output values. You can use table output values to output the contents of table cells. For some types of tables, you can specify a row range from which to choose the table cells. During the run session, UFT One retrieves the current data from the specified table cells according to the settings that you specified and outputs the values to the Data pane.



Select the values to add in datatable and verify in the script.

### Text and Text Area Output Values

You can use text output values to output text strings displayed in an application. When creating a text output value, you can output a part of the object's text. You can also specify the text before and after the output text.

You can use text area output values to output text strings displayed within a defined area of a screen in a Windows-based application.

Test Checkpoint can capture data before and after texts

Example:

Try for Order number

### Database Output value

You can use database output values to output the value of the contents of database cells, based on the results of a query (result set) that you define on a database. You can create output values from the entire contents of the result set, or from a part of it. During the run session, UFT One retrieves the current data from the database and outputs the values according to the settings that you specified.

We can store data from database to global sheet to retrieve the value

### XML Output value

You can use XML output values to capture and output the values of XML elements and attributes in XML documents.

For example, suppose that an XML document in a Web page contains a price list for new cars. You can output the price of a particular car by selecting the appropriate XML element value to output.