

# What is QTP/UFT?

**QTP** is an automated functional [Testing](#) tool that helps testers to execute automated tests in order to identify any errors, defects or gaps in contrary to the expected results of the application under test. It was designed by Mercury Interactive and later on acquired by HP and now MicroFocus. Full form of QTP is QuickTest Professional while UFT means Unified Functional Testing.

## Advantages of QTP Automation

- It supports record and playback
- It uses an active screen to record scripts and helps tester in referring the screen object properties
- It has excellent object identification process or mechanism
- It supports different add-ins like Oracle, Java, SAP, NET, Web Forms, People soft, etc..
- It allows you to enhance the existing tests even without the AUT through an active screen
- It supports popular automation frameworks- keyword driven testing approach, modular testing approach, data-driven testing approach, etc..
- It comes with an inbuilt IDE
- It can be integrated with Test management tools like Quality Center, Test Director, and Winrunner
- Different types of suites like Smoke, Regression, Sanity can be easily maintained
- It supports XML
- Test reporting is possible through QTP for analysis purpose
- Easy to maintain

## Features of the latest version of QTP/UFT

The latest version of QTP/UFT (Unified Functional Testing) includes new features

New features	Details
<ul style="list-style-type: none"> <li>Support for OS and browser</li> </ul>	<ul style="list-style-type: none"> <li>Supports Windows 8.1 and Windows server 2012</li> <li>Performs testing on Safari browser</li> </ul>
<ul style="list-style-type: none"> <li>UFT extension in Chrome store</li> </ul>	<ul style="list-style-type: none"> <li>Receive updates automatically on chrome</li> </ul>
<ul style="list-style-type: none"> <li>Support for windows runtime objects</li> </ul>	<ul style="list-style-type: none"> <li>Record and run tests on your Windows Runtime application</li> <li>Create object repositories with Windows Runtime application</li> </ul>
<ul style="list-style-type: none"> <li>Newly supported technologies</li> </ul>	<ul style="list-style-type: none"> <li>JDK 1.8</li> <li>XenDesktop 7</li> <li>SAP Web Dynpro ABAP for NetWeaver 7.40 and so on</li> </ul>
<ul style="list-style-type: none"> <li>Extended support for various technologies</li> </ul>	<ul style="list-style-type: none"> <li>Recognizes and interact with objects from Siebel web application etc..</li> <li>New FlexTable test object methods available for advanced data</li> </ul>

## What is QTP Parameterization?

QTP provides us to pick different test inputs at a run time. This process of providing different input values through external parameters is called as parameterization.

## Types of parameterization in QTP

Parameter types can be:

1. Data Table parameters
2. Test/Action parameters
3. Environment variable parameters
4. Random number parameters

## What is Checkpoint in QTP?

A checkpoint is a verification point that compares the current value with the expected value for specified properties of an Object. If the current and expected value match it generates a PASS status otherwise FAIL status.

## Types of Checkpoints in Micro Focus UFT

There are many types of checkpoints detailed below

1. **Standard checkpoints:** It compares the expected values of object properties captured during recording to the **object's current values** during a run session
2. **Page Checkpoint:** A Standard Checkpoint created for a web page can be called a Page Checkpoint. It is used to check a total number of links & images on a web page. Page Checkpoints can be used to check Load Time i.e. time taken to load a web page.
3. **Bitmap Checkpoint** helps a user in checking the bitmap of an image or a full web page. It does a pixel by pixel comparison between actual and expected images.
4. **Image Checkpoint** enables you to check properties like source file location of a web image. Unlike, Bitmap Checkpoint you can not check pixels(bitmaps) using image checkpoint.
5. **Text Checkpoint** is Used to check expected text in a web-page or application. This text could be from a specific region of the application or a small portion of text displayed
6. **Accessibility Checkpoints** verifies compliance with World Wide Web Consortium (W3C) instructions and guidelines for Web-based technology and information systems. These Guidelines make it easy for disabled to access the web.

7. **Database Checkpoints** create a query during record time and database values are stored as expected values. The same query is executed during run time and actual & expected values are compared.
8. In **Table Checkpoint**, you dynamically can check the contents of cells of a table (grid) appearing in your environment. You can also check various table properties like row height, cell width and so on. Table Checkpoint is similar to Database Checkpoint
9. Using **XML Checkpoints** you can verify XML Data ,XML Schema, XML Data

## What is Action in QTP?

**Actions** help divide your test into “logical units” or “Business Processes”. Actions help create a script which is more modular and efficient.

When a script is newly created it consists of only one action. But you can add more Actions to your Micro Focus UFT Script as per requirements.

There two types of Actions

1. **Reusable** Actions: can be used in other Tests. They can be used in the same Test Script multiple times.
2. **Non-Reusable** Actions: cannot be used in other Tests. They can be called in the same script only once

At times, if an action becomes big, it's a nice practice to split it. You can split an existing action in two ways

1. **Independent** of each other which splits the selected action into two sibling actions
2. **Nested Action** which splits the selected action into a parent action whose last step calls the second, child action

Actions can access test-data stored in datasheets. HP QTP provides **2 type of datasheets**

1. **GLOBAL Datasheet:** It is Unique for the entire test. Any Action can access and write data into Global Datasheet. A sheet is named “GLOBAL”
2. **LOCAL Datasheet:** Equal to number of Actions in the sheet. An Action can read and write data into its own local Datasheet only. Sheet name = “ACTION NAME”.

## What is Object Repository in QTP?

Object Repository is a collection of Test Objects and information that is recognized by QTP for working on it. When a user records a test, the objects and their properties are captured by default.

## Types of Object Repository

There are 2 Types of Object Repository in QTP

1. Local Object Repository
2. Shared Object Repository

## Local Object Repository

- Local Object Repository is the default object repository
- It is specific to actions and can be used only for a particular action
- Local Object Repository is preferable when the application is not dynamic with respect to time
- Local Object Repository cannot be reused
- You can perform many operations in the local object repository such as –
- Highlight an object stored in a repository on the application under test

- Check whether a particular object in your AUT is stored in the Object Repository
- Cut, Copy, Paste, Modify and Delete Objects
- In case you have accidentally modified the value of a property you can update its description from the application using update function

## **Shared Object Repository: Create, Associate, Edit**

- Global or Shared Object Repository is preferable when an application is dynamic and object description change frequently
- Between Shared and local object repository, shared object repository is more commonly used in automation projects
- However, it has maintenance and administration overheads as compared to local object repository.

To create and use a shared object repository you need to perform three broad steps

- Creating a Shared Object Repository
- Associating a Shared Object Repository
- Editing a Shared Object Repository

Let's look at them one at a time

### **Step 1) Creating a Shared Object Repository**

- All repositories are local by default. To create a Shared Object Repository, in the Object Repository Dialog Box, Click File > Export Local Objects
- Repository files have an extension .tsr .Give a suitable name say “guru99” and save
- The Shared Repository File is now created

### **Step 2) Associating a Shared Object Repository**

- Next step is to associate the repository to your test, which enables you to use it
- To associate a repository with a test, Click Resources > Associate Repository
- You can select the Repository to associate with Actions available in your test.
- Now you can now use this shared repository to develop your test

### **Step 3) Editing a Shared Object Repository**

- You can use the Object Repository Manager to Edit a Share Repository.
- Select Resources > Object Repository Manager. Open the Object Repository we created “guru99”
- By Default, Repository is opened in Read-only mode. To enable editing click File > Enable Editing
- Once editing is enabled you can all the operations like cut, copy, paste, rename objects etc that you can also do in Object Repository
- Using the Object Repository Manager is you can compare two Object Repositories. QTP will give you a static’s of what’s unique and common in both the repositories
- You can use the Object repository merge tool to merge two repositories into one

### **Recording Mode**

HP UFT (QTP) supports 3 types of recording modes

1. Context Sensitive
2. Analog
3. Low Level

## Context Sensitive Recording mode

- The normal recording mode is also called Context Sensitive Mode
- It is the default mode of recording which takes full advantage of Quick Test Professional's test object model.
- It recognizes objects in the application regardless of their location on the screen.
- It records the objects in your application and the operations performed on them

## Analog Recording Mode

- In an analog recording mode, Quick Test Professional records and tracks every movement of the mouse as you drag the mouse around a screen or window.
- Micro Focus UFT's Analog recording is captured as Tracks and stored in the directory of your test
- It is useful for recording operations that cannot be recorded at the level of an object. Eg., A signature produced by dragging the mouse
- In Analog mode you can record
  1. Record Relative to screen
  2. Relative to window
- When your analog operation is confined to just one window, use relative to a window
- When your analog operation involves multiple screens like dragging and dropping an object from one window to other use the screen option



## Low-Level Mode

- This mode enables you to record on any object in your application, irrespective of QTP recognizes the specific object or the specific operation.
- This mode records at the object level and records all run-time objects as either Window or WinObject test objects.
- It is used when the exact coordinates of the object are important for your tests. A good example would be hashmaps where clicking different sections of a picture take you to different links
- Used when recording tests in an environment (or on an object) not recognized by QTP
- Low-level mode records the x,y coordinates of any clicks

## Functions

if you have segments of code that you need to use several times in your tests, you may want to create a user-defined function. By using user-defined functions, your tests are shorter, and easier to design, read, and maintain.

## Transactions

n Micro Focus UFT/QTP, You can measure how long it takes to run a section of your test by defining **Transactions**. You define transactions within your test by enclosing the appropriate sections of the test with start and end transaction statements.

Transactions can be inserted anywhere in the script

There is no limit to the number of transactions that can be added to a test.

## OBJECT SPY

object Spy can help determine the useful properties and methods associated with an object in your environment.

The HP/Micro Focus UFT tutorials also describes **GetROProperty, GetTOProperty & SetTOProperty**

### GetRoProperty

- GetRoProperty – is an inbuilt method used to retrieve the runtime value of an object property.
- There are 4 steps involved in using the GetRoProperty
- Step 1) Record the Object on which you want to use the GetRoProperty in Object Repository
- Step 2) For the recorded object identifies the run-time property which could be used. You may use Object Spy.
- Step 3) Use GetRoProperty method to retrieve the identified Run-time property and store the value in a variable
- Step 4) Use this value for further deductions

### SetToProperty & GetToProperty

- Consider a Web Button stored in the Object Repository
- When the test is run QTP creates a copy of this object called Test Object and compares it with the Run Time Object
- Using GetToProperty you can retrieve the value of a property of Test Object
- **Using SetToProperty you can change** the property value of a Test Object
- When the test is completed, this test object is discarded and so are any modifications you made in the object properties using the SetToProperty
- When the test is re-run, a new copy of the test object is created with original property values stored in the object repository

- You can consider using `GetToProperty` and `SetToProperty` when your test script has multiple lines of codes and your environment is sporadic
- For a note, there is no `SetRoProperty`