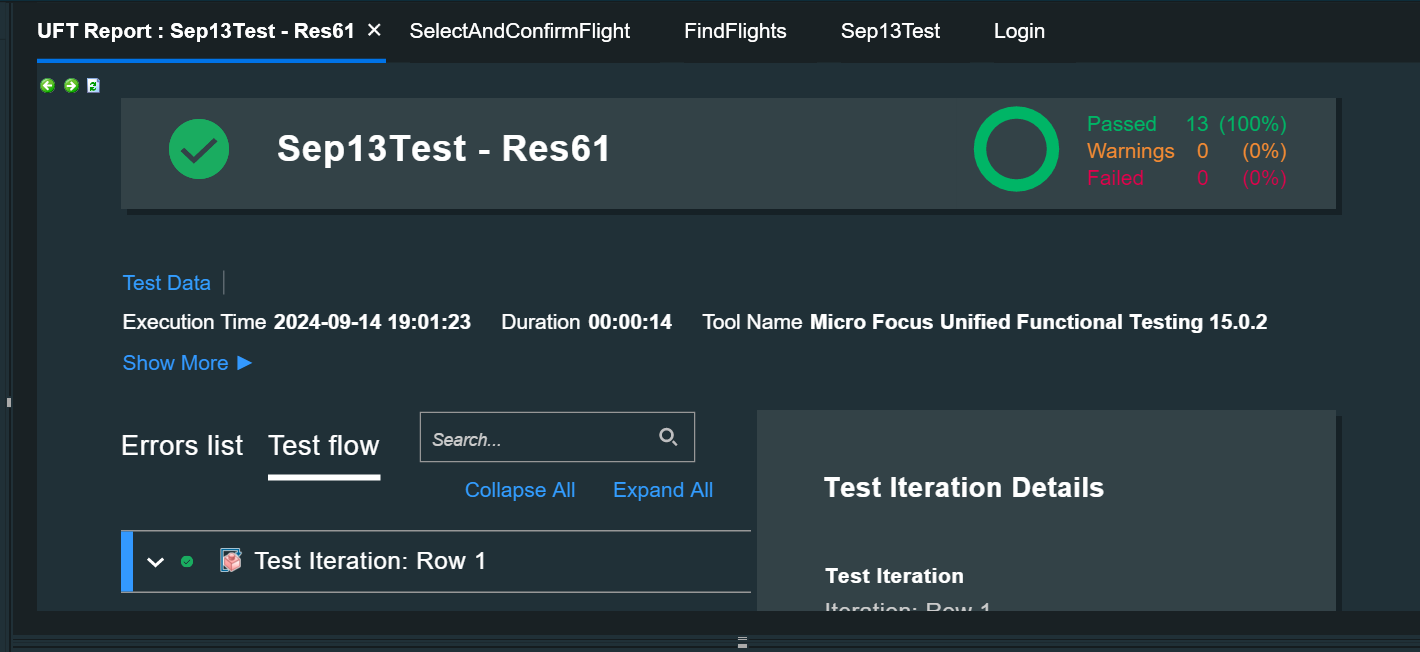
UFT Test Script Creation

# Record and Playback

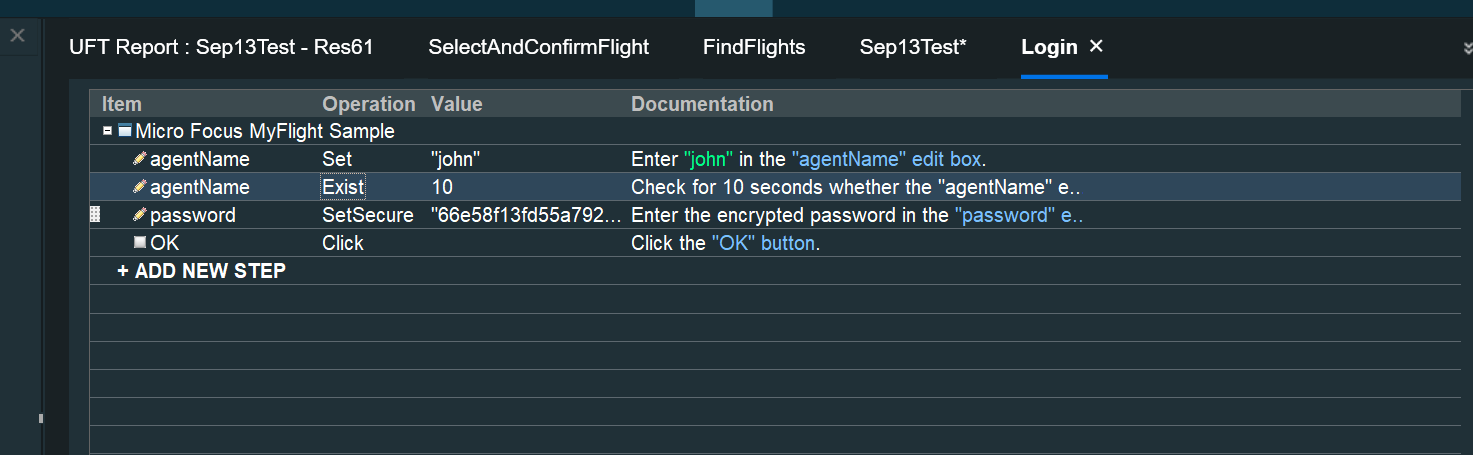
1. Click the toolbar button arrow, and select Add New Test.
2. Select to add a GUI Test to the solution, and enter the test name as MyFlight.
3. Leave the Location with the default value, and then click Add. A blank test opens, showing the MyFlight test flow in the canvas, and another tab for a blank action, named Action1.
4. In the canvas, right-click the Action1 box, and select Action Properties. In the Name field, enter Login.
5. Similarly add actions for flight search and booking.

**Test Result:**



# Keyword View and Expert View

In Unified Functional Testing (UFT), the Keyword View is a feature that lets users create and view test steps in a table-like format. The Keyword View is useful for non-programmers because it allows them to create test steps without writing code.



# Active Screen

The Active Screen pane enables you to view snapshots of your application as it appeared during a step in a recording session.

Right-click in the Active Screen to add steps, checkpoints, or output values after running your test, without opening your application.

## Open Active Screen

Select View > Active Screen.

## Change Active Screen

1. Ensure that the updated window or page in your application is open and available, and that the Active Screen pane is displayed in UFT One.
2. In the Keyword View, click a step that you want to change. The window or page is displayed in the Active Screen pane.
3. Select **Tools > Change Active Screen**. The UFT One window is hidden and the mouse pointer becomes a pointing hand.
4. Click the window or page displayed in your application.
5. When a message prompts you to change your current Active Screen display, click **Yes**.

## Enabling/Disabling Active Screen Capture

Select Tools > Options > GUI Testing tab > Active Screen node > Custom Level.

### Capture Levels

1. **Complete** - Instructs UFT One to save all description properties of all objects in the application
2. **Partial (Default)** - All properties of the recorded object only, in subsequent steps in the same window
3. **Medium** - Instructs UFT One to save all description properties for the recorded object plus all description properties for the parent objects in the recording hierarchy
4. **None** – Disable active screen capture

# Object Repository

In UFT One, the object repository is a storehouse for objects used in GUI tests and components. It stores test objects that correspond to objects learned in an application.

* Adding Object to Repository and using in the test
* Rename and highlight the object

WpfWindow("Micro Focus MyFlight Sample").WpfEdit("agentName").Set "john"

WpfWindow("Micro Focus MyFlight Sample").WpfEdit("agentName").Exist 10

WpfWindow("Micro Focus MyFlight Sample").WpfEdit("password").SetSecure "66e58f13fd55a792852e"

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

if WpfWindow("HPE MyFlight Sample Applicatio").WpfTabStrip("BookFlight").Exist then

Reporter.ReportEvent micPass, "Login", "Login Successful"

else

Reporter.ReportEvent micFail, "Login","Login Failed"

End If

Note: - If object is not found in Object Repository (OR), UFT is unable to identify the object in AUT

## Local and Shared Object Repository

In UFT One, local and shared object repositories are two types of object repositories that can store test objects:

* Local object repository
* Shared Object Repository

### Local Repository

A file that stores objects for a specific action or component. A local object repository is created automatically when a new action or component is created.

### Shared Repository

A file that stores objects that can be accessed by multiple actions or components. Shared object repositories are opened as read-only by default.

You can use a combination of local and shared object repositories, or store objects in either type of repository exclusively.

**Creating a Shared Object Repository**

1. Repositories are created using Object Repository Manager
2. Add all the required objects to Manager
3. Save the Repository (Extension will be .tsr)
4. Associate the repository with actions
5. Use in the script

## Object Repository Vs Object Repository Manager

You perform many object repository-related tasks either in the Object Repository window or in the Object Repository Manager. Some object repository-related tasks can also be performed in both.

|  |  |  |
| --- | --- | --- |
| Features | Object Repository | Object Repository Manager |
| Adding and deleting test objects | Yes | Yes |
| Highlighting a test object in your application | Yes | Yes |
| Locating a test object in the object repository | Yes | Yes |
| Specifying or modifying object property values | Yes | Yes |
| Updating object property values directly from objects in your application | Yes | Yes |
| Renaming test objects | Yes | Yes |
| Exporting local objects to a shared object repository | Yes | No |
| Merging multiple object repositories | No | Yes |

## Merging Repository

How to merge two repositories?

* Login
* Book Flight

Resources 🡪 Object Repo Manager 🡪 Tools 🡪 Object repository merge tool

# Object Spy

The Object Spy in UFT One lets you see the properties and operations of an object in an open application. You can also view the test object hierarchy, description properties, and operations that UFT One uses to represent the object.

To access the Object Spy, do one of the following:

* Click the down arrow near the **Object Identification Center**  toolbar button and select the **Object Spy** .
* In the Record toolbar, click the **Object Spy** button during a recording session.
* In the Object Repository window or Object Repository Manager, click the **Object Spy** toolbar button .

## Spy on your object

Spy on an object in your application to view or capture properties and/or operations for that object.

To spy on an object:

* **If you want to select the object by clicking it,** simply click the pointing hand button , and select the object in your application.
* **If you want to select the object by hovering over it instead of clicking**, first turn on the **Hover mode** above the toolbar. Then click the pointing hand  to switch to your application and hover over your object.

## Add objects to the object repository

To use your spied object in GUI tests and components, first add it to an associated object repository.

To add objects to the object repository:

1. Spy on an object in your application, and select it in the Object Spy **Object hierarchy.**
2. In the Object Spy toolbar, click the **Add to Repository**button .

# Object Identification Center

The Object Identification Center (OIC) is a next-generation spy tool that enables you to create unique and robust descriptions for any object in an open application.

As the default spy, the OIC lets you spy on objects in your application, determine how to identify those objects, and add them to your tests and object repositories.

Use the OIC to:

* Spy on single or multiple objects
* Create customized object descriptions
* Add single or multiple objects to object repositories
* Drag single or multiple objects into your test to create steps
* Copy object descriptions to use in programmatic descriptions
* Compare the descriptions of two objects
* Capture images of objects in an application

## Spying the object using OIC

* **Hover mode highlight**: Enable this option to highlight objects you point at when spying in hover mode.
* **Hover mode Time Out**: Specify how many seconds to hover over an object to capture it in the OIC. The default value is 1.3 seconds.
* If you are spying in hover mode, hover over the object you want to capture.
* In single object spy mode, when you click an object, the OIC spies on it and directly displays the object properties in edit mode.
* To make sure that you have the object that you intended, in the OIC window, hover over the object, and click the **Highlight** button  to highlight it.
* T o remove multiple objects from the list, click the **Clear all**button  first, then select their rows and click the **Delete** button .
* To start a new spying session, click the **Delete**button  to delete all objects, and then click the **Start spying** button .

# Comaping two Object Repositories

Select the Shared Object Repositories to compare

1. In the Object Repository Manager window, select **Tools > Object Repository Comparison Tool**.
2. In the New Comparison dialog box, specify the two object repository files you want to compare.

Analyze the initial comparison results

After the comparison is complete, view the results summary in the Comparison Statistics Dialog Box.

Analyze the detailed comparison results

Review and analyze the comparisons between the repositories in the Object Repository Comparison Tool Main Window.

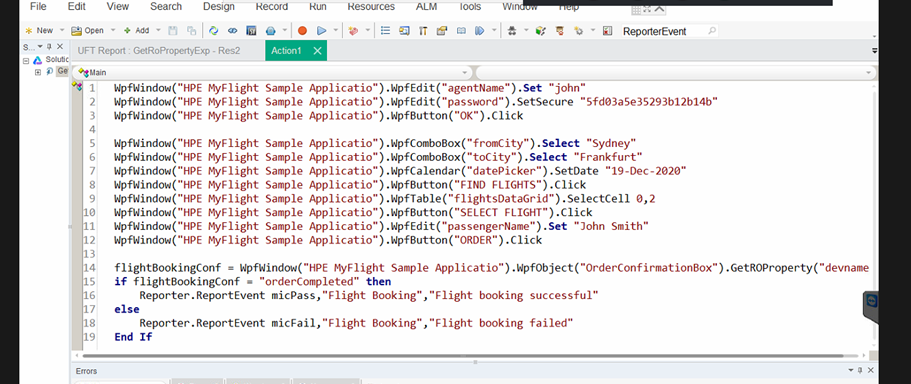
# GetROProperty

GetROProperty 🡪 Get Runtime Object Property

Manual test case:

1. Login to flight reservation application
2. Enter all mandatory fields
3. Click on Create order

Expected result: Order number is generated

****

# Actions – Types

In a GUI test, each test is comprised of **actions**. An action is a separate modular test script, including all of the steps in that action, and any objects in its local object repository and any associated shared object repositories.

The actions used in the test, and the order in which they are run, are displayed in the canvas.

Actions is divided into:

* **Reusable actions**
* Default type.
* Can be called multiple times by the local test and other tests.
* Must be updated from the original test.
* Can be marked as non-reusable to change its type.
* **Non Reusable Actions**
* Can be called only once, and in the local test.
* Can be copied.
* Can be marked as reusable to change its type.

## Calling a reusable action

You can either call the action from the canvas or in the script by using the below command

RunAction "Login", oneIteration

Using SystemUtil.run

SystemUtil.Run "C:\Program Files (x86)\HPE\Unified Functional Testing\samples\Flights Application\FlightsGUI.exe"

### Types of Action calls

* **Call to New Action** – Creates a new action which will be called after the current action
* **Call to Copy of Action** - When you make a Copy of an Action, the action is copied in its entirety, including checkpoints, parameterization, and the corresponding action tab in the Data Table into the calling test. When you insert a copy of an existing action, you can make changes to the copied action, and your changes will not affect nor be affected by any other test.
* **Call to Existing Action** - Calls to actions are read-only in the calling test. They can only be modified in the test in which they were created. Enables you to use the same action in several tests and makes it easy to maintain tests.