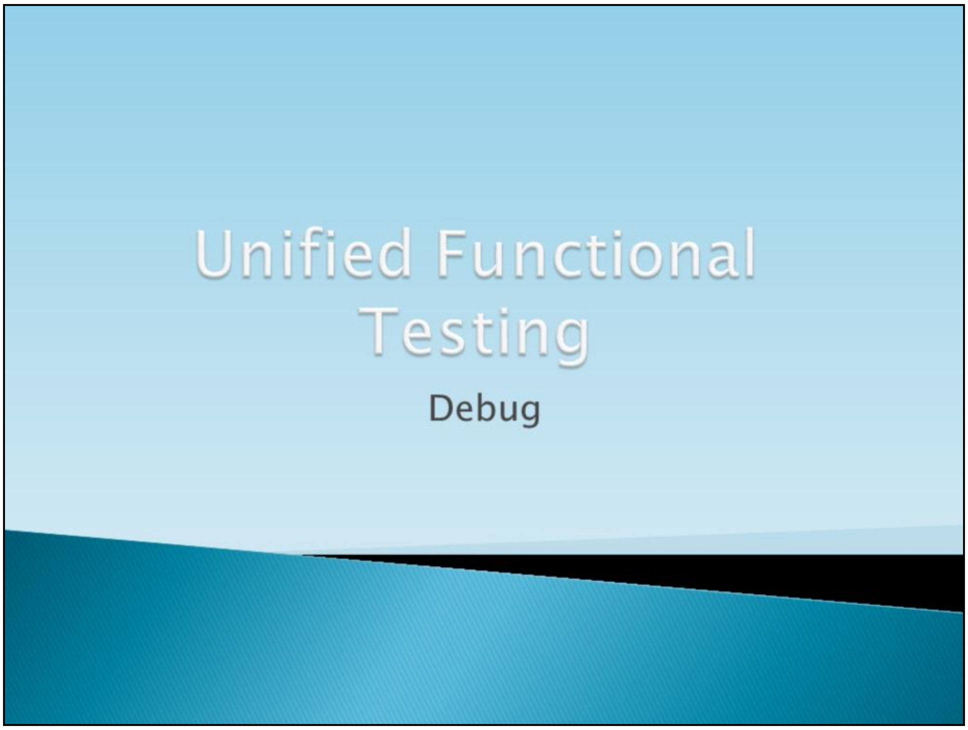
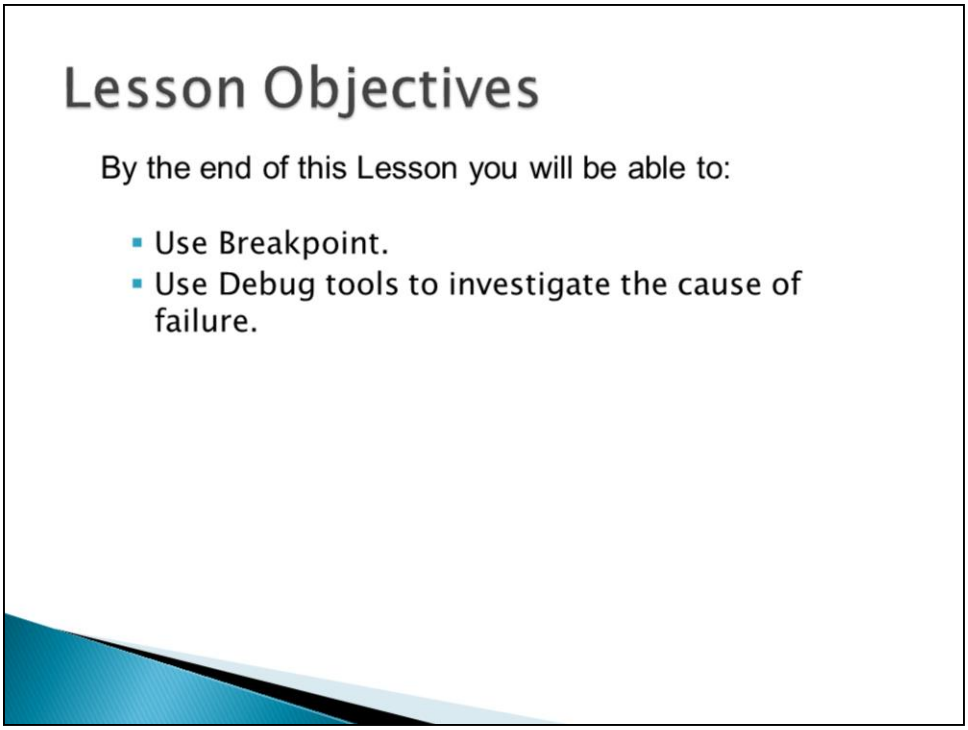


Unified Functional Testing

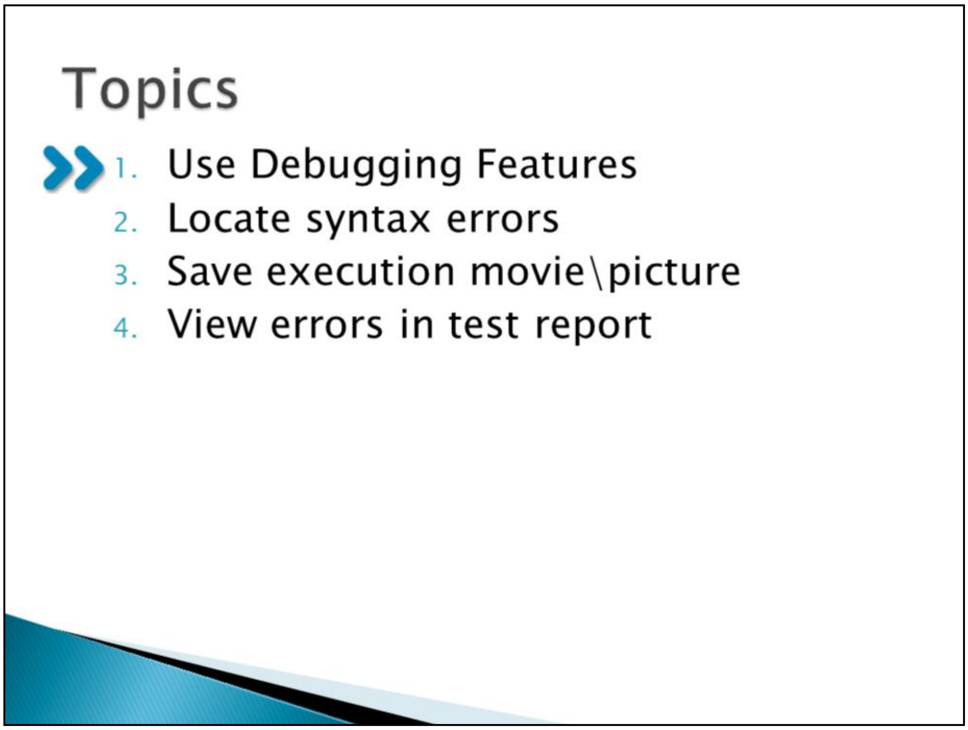
Debug



Lesson Objectives

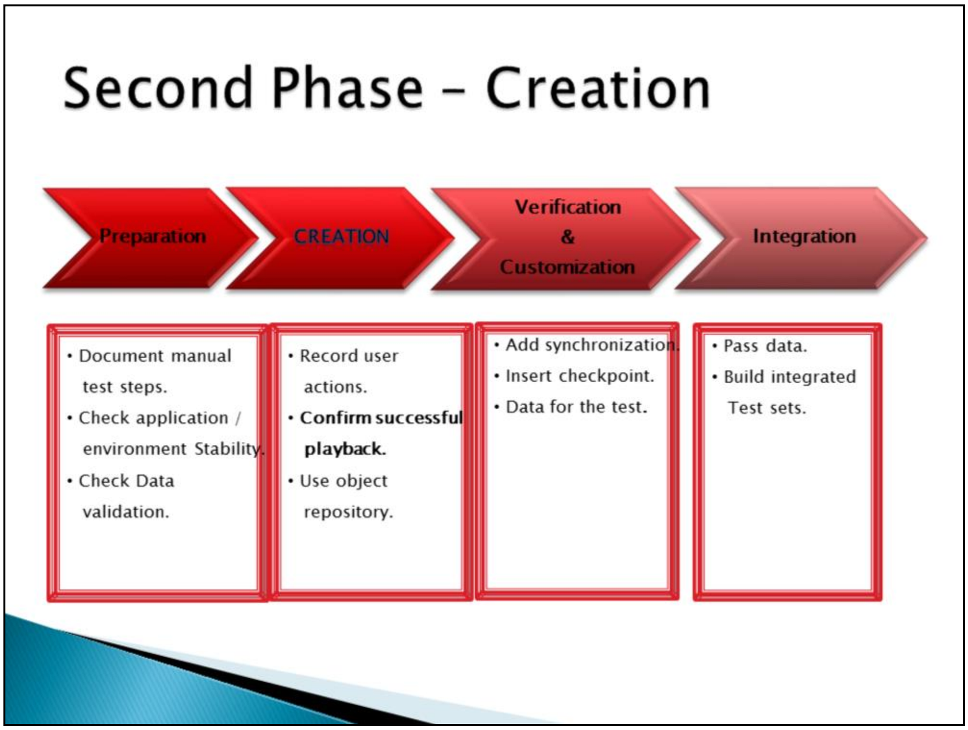
By the end of this Lesson you will be able to:

- Use Breakpoint.
- Use Debug tools to investigate the cause of failure.



Topics

1. Use Debugging Features
2. Locate syntax errors
3. Save execution movie\picture
4. View errors in test report



Second Phase – Creation

Preparation

- Document manual test steps.
- Check application / environment Stability.
- Check Data validation.

CREATION

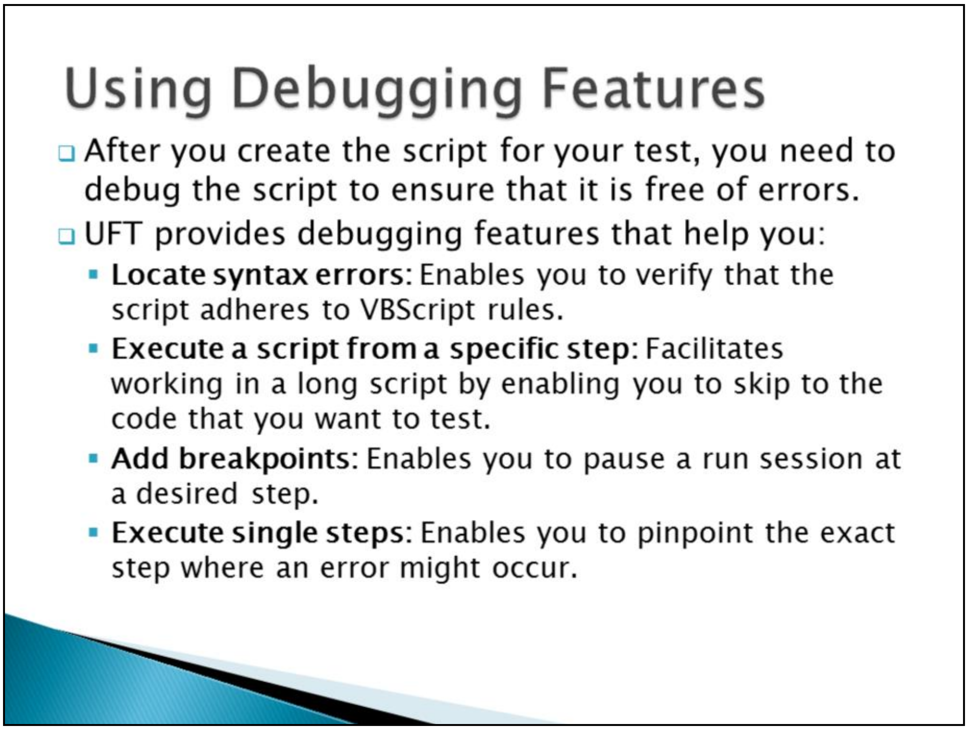
- Record user actions.
- **Confirm successful playback.**
- Use object repository.

Verification & Customization

- Add synchronization.
- Insert checkpoint.
- Data for the test.

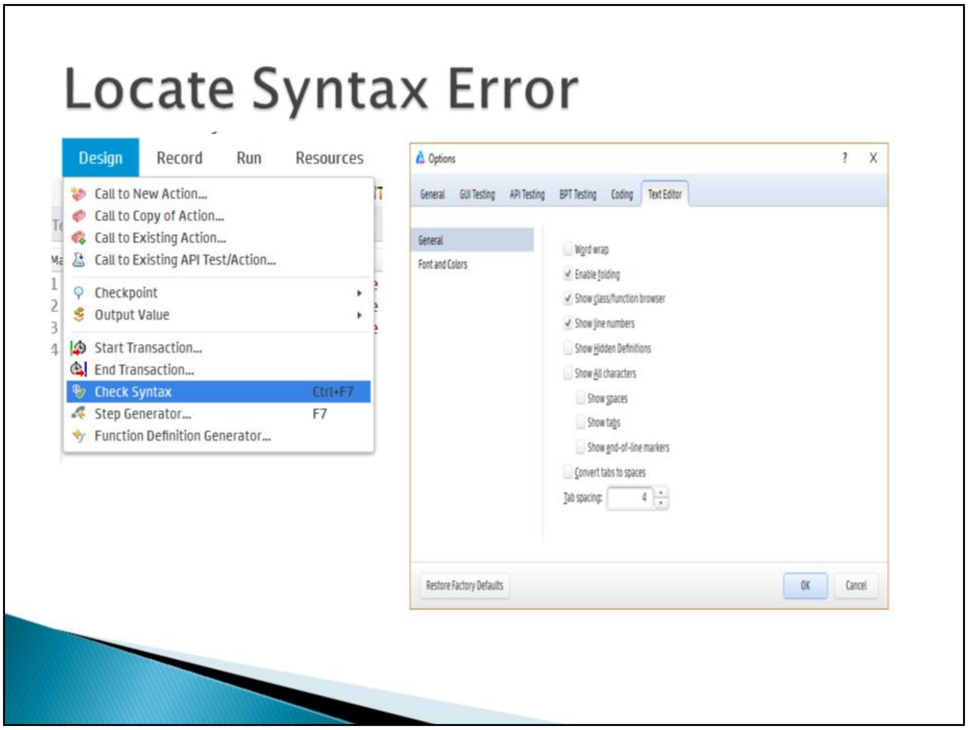
Integration

- Pass data.
- Build integrated Test sets.



Using Debugging Features

- After you create the script for your test, you need to debug the script to ensure that it is free of errors.
- UFT provides debugging features that help you:
  - **Locate syntax errors**: Enables you to verify that the script adheres to VBScript rules.
  - **Execute a script from a specific step**: Facilitates working in a long script by enabling you to skip to the code that you want to test.
  - **Add breakpoints**: Enables you to pause a run session at a desired step.
  - **Execute single steps**: Enables you to pinpoint the exact step where an error might occur.



Locate Syntax Error

□ Word wrap
☑ Enable folding
☑ Show class/function browser
☑ Show line numbers
□ Show Hidden Definitions
□ Show All characters
□ Show spaces
□ Show tabs
□ Show end-of-line markers
□ Convert tabs to spaces
Tab spacing: 4

You can perform a syntax check from the **DESIGN→CHECK SYNTAX** menu.

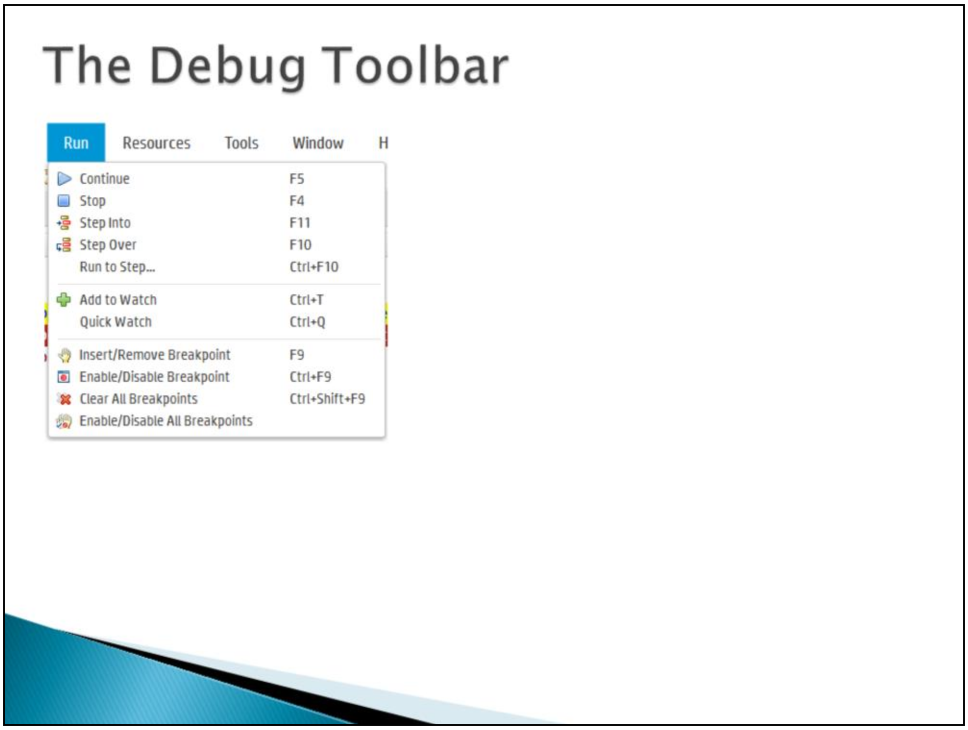
For each error found in your **script,** the **ERRORS** pane displays the following information:

•**DESCRIPTION**: Provides the description of the error. For example, if you open a conditional block with an IF statement but do not close it with an END IF statement, the description of the error is: EXPECTED 'END IF'.

•**TEST:** Provides the name of the test or function library that contains with the syntax error.

•**ITEM:** Provides the name of the action that contains the syntax error.

•**LINE:** Provides the line number that contains the syntax error. To display all the line numbers in a script, from the UFT menu bar, select **TOOLS →OPTIONS→TEXT EDITOR,** and check the **SHOW LINE NUMBERS** check box in **GENERAL** dialog box.



The Debug Toolbar

To view the DEBUG toolbar, from the UFT menu bar, select **VIEW →TOOLBARS → DEBUG.** The DEBUG toolbar consists of the following buttons:

• **PAUSE:** Temporarily suspends a test. A paused test stops running when all previously interpreted steps are executed.

• **STEP INTO:** Runs the current step of the active test. If the current line calls another action, the called action is displayed. The test pauses at the first line of the called action.

• **STEP OVER:** Runs the current step in the active test. When the current step calls another action, the called action is executed but not displayed.

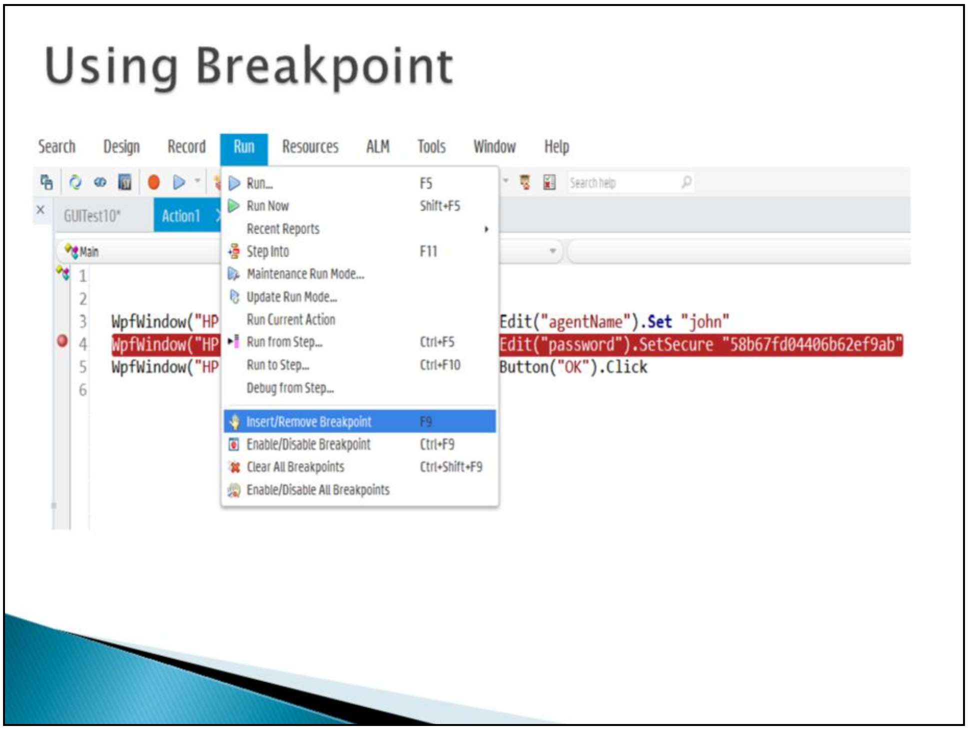
• **STEP OUT:** Runs the current step until the end of the function, returns to the calling component or function library, and then pauses the run session.

• **INSERT/REMOVE BREAKPOINT:** Stops a test run at a specific step in the test. A red icon appears in the left margin of the step to indicate that a breakpoint is inserted on the step. The test runs until it reaches the breakpoint. You can examine or manipulate steps up to the breakpoint. To remove a breakpoint, click

INSERT/REMOVE BREAKPOINT again.

• **CLEAR ALL BREAKPOINTS:** Removes all breakpoints within a test.

• **ENABLE/DISABLE ALL BREAKPOINTS:** Enables or disables all breakpoints within a test.



Using Breakpoint

```
WpfWindow("HP                                   Edit("agentName").Set "john"
WpfWindow("HP                                   Edit("password").SetSecure "58b67fd04406b62ef9ab"
WpfWindow("HP                                   Button("OK").Click
```

Breakpoints enable you to suspend a run session at a predetermined point in the script so that you can inspect the run-time state of your application. To set a breakpoint:

1. Place the cursor on the step where you want to add the breakpoint.

2. From the UFT menu bar, select **RUN→INSERT/REMOVE BREAKPOINT.** The Breakpoint symbol appears on the left margin of the script.

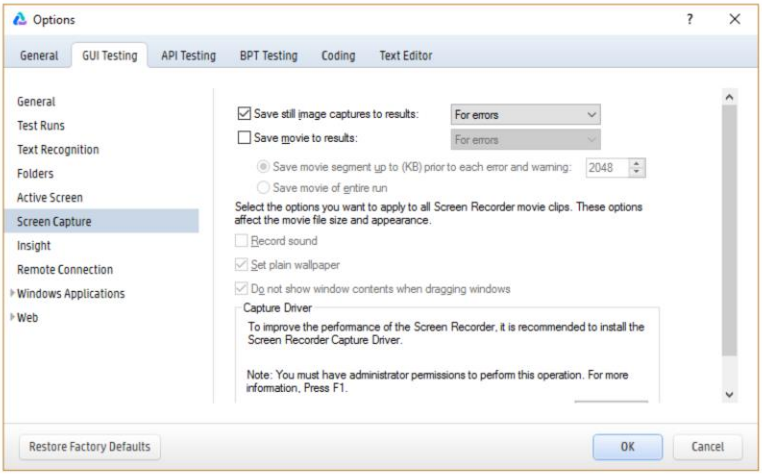
3. UFT pauses the run session when it reaches the breakpoint before executing the step.

4. To remove a breakpoint, place the cursor on the step and from the UFT menu bar, select **RUN→INSERT/REMOVE BREAKPOINT** again**.**

5. When you complete debugging, you can clear all the breakpoints from a script. To remove all the breakpoints, from the UFT menu bar, select **DEBUG → CLEAR ALL BREAKPOINTS.**

Note: You can also insert a breakpoint by clicking next to the line number of the step where you want breakpoint.

Still Images and Movies



☑ Save still image captures to results: For errors

□ Save movie to results: For errors

● Save movie segment up to (KB) prior to each error and warning: 2048

○ Save movie of entire run

Select the options you want to apply to all Screen Recorder movie clips. These options affect the movie file size and appearance.

□ Record sound

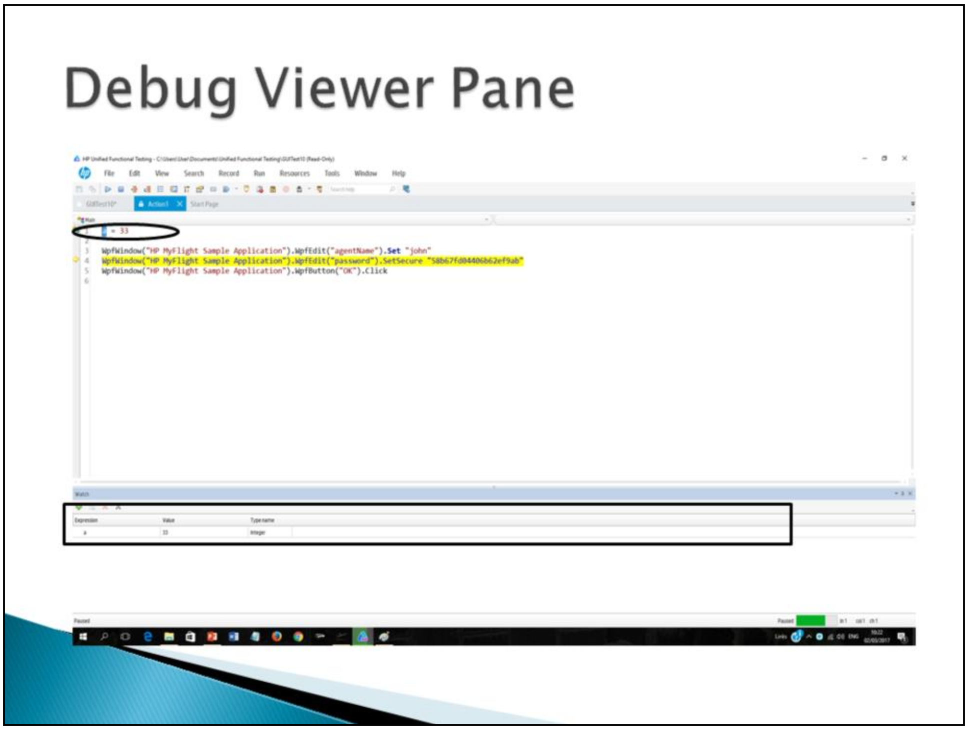
☑ Set plain wallpaper

☑ Do not show window contents when dragging windows

To improve the performance of the Screen Recorder, it is recommended to install the Screen Recorder Capture Driver.

Note: You must have administrator permissions to perform this operation. For more information, Press F1.

Debugging your test may be easier if you configure the run option to capture and save the step screen, irrespective of the test result. This enables you to analyze the complete test run.



Debug Viewer Pane

HP Unified Functional Testing - C:\Users\User\Documents\Unified Functional Testing\GUITest10 (Read-Only)
File
Edit
View
Search
Record
Run
Resources
Tools
Window
Help
GUITest10*
Action1
Start Page
Main
Watch
Paused

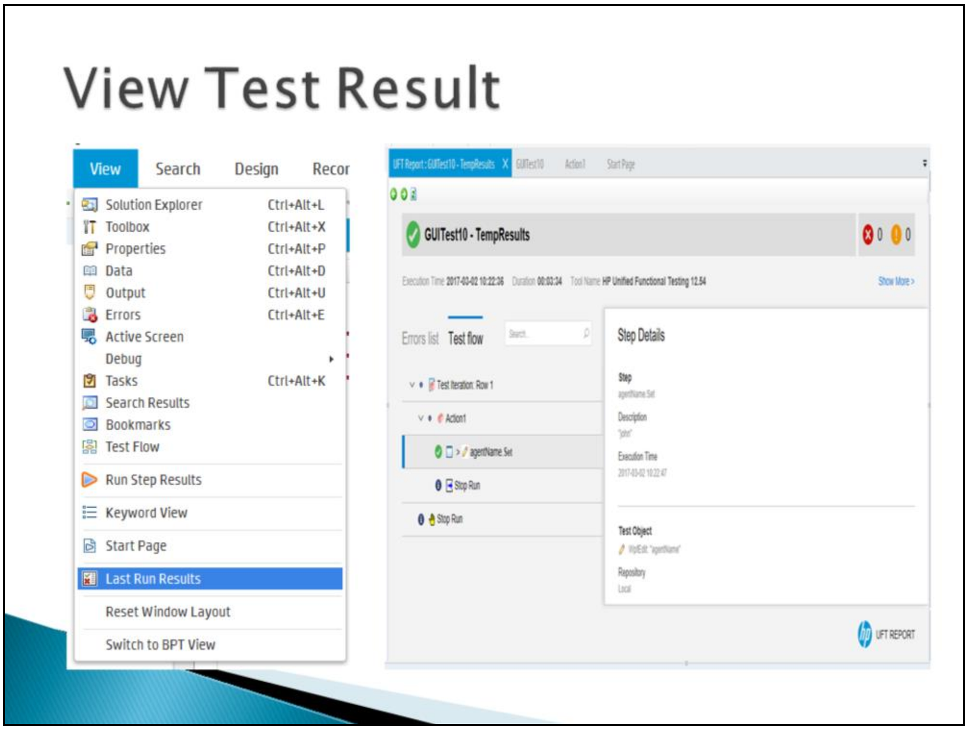
```
a = 33

WpfWindow("HP MyFlight Sample Application").WpfEdit("agentName").Set "john"
WpfWindow("HP MyFlight Sample Application").WpfEdit("password").SetSecure "58b67fd04406b62ef9ab"
WpfWindow("HP MyFlight Sample Application").WpfButton("OK").Click
```

Expression	Value	Type name
a	33	Integer

The Debug Viewer tabs are used to display the values of variables and objects in the main script of the current action, or in a selected subroutine.

Debug Viewer contain 3 tabs:

1. **Watch** - The Watch tab enables you to view the current value of any variable or VBScript expression that you added to the Watch tab.
2. **Local Variables -** During a run session, the Variables tab displays the current value of all variables that have been recognized up to the last step performed in the run session.
3. **Console -** The Console tab enables you to run a line of script to set or modify the current value of a variable or VBScript object in your test or function library. When you continue the run session, UFT uses the new value that was set in the command.



View Test Result

Execution Time 2017-03-02 10:22:36 Duration 00:03:34 Tool Name HP Unified Functional Testing 12.54

Step
agentName.Set

Description
"john"

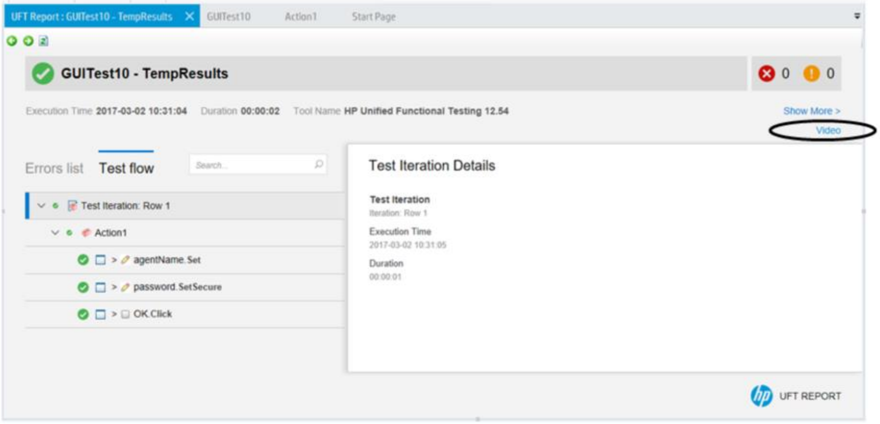
Execution Time
2017-03-02 10:22:47

Test Object
WpfEdit: "agentName"

Repository
Local

After you run your test, you can view the results in the TEST RESULTS window.

Screen Recorder



GUITest10 - TempResults ✕ 0 ! 0

Execution Time 2017-03-02 10:31:04 Duration 00:00:02 Tool Name HP Unified Functional Testing 12.54

- Test Iteration: Row 1
  - Action1
    - ✓ agentName.Set
    - ✓ password.SetSecure
    - ✓ OK.Click

Test Iteration Details

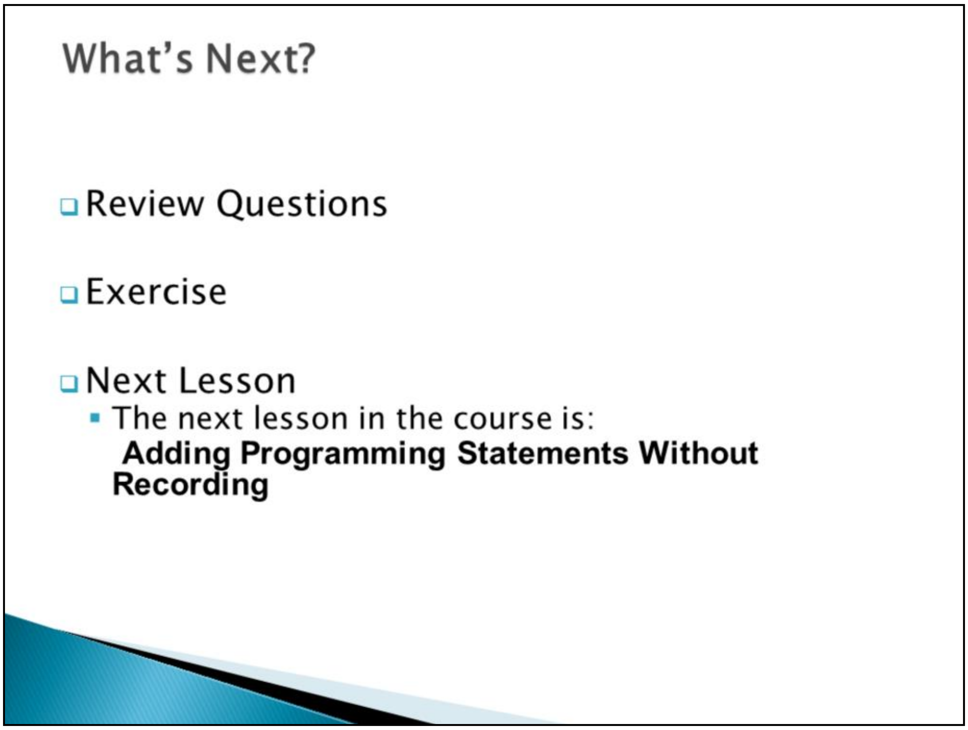
Test Iteration
Iteration: Row 1

Execution Time
2017-03-02 10:31:05

Duration
00:00:01

The Mercury Screen Recorder saves a movie of the test run. You may save a movie of the entire test run or only segments surrounding errors and warnings.

The checkbox in SCREEN CAPTURE window must be selected to save a movie of your test run.



What’s Next?

- Review Questions
- Exercise
- Next Lesson
  - The next lesson in the course is:
    Adding Programming Statements Without Recording

End of Lesson