



vivit

Discover the independent Micro Focus software user community where you can share, collaborate, exchange, and grow



COMMUNITY



**Building a Reusable Automated Testing Framework for UFT**  
**January 23, 2019**



# Sponsored By

## Vivit's Testing Quality and ALM Special Interest Group



Andreas Birk  
Founder and Principal  
Consultant of SWPM  
Software Process Management



Mihai Grigorescu  
Micro Focus Testing  
Practice Director  
Accenture



Robert Linton  
Business Consultant,  
ADM  
Micro Focus  
Government Solutions  
(MFGS)



Christopher Scharer  
Sr. SQA Test Automation  
Analyst  
Berkley Technology Services



# Hosted By



Robert Linton  
Business Consultant, ADM  
Micro Focus Government Solutions (MFGS)

YOUR INDEPENDENT MICRO FOCUS SOFTWARE USER COMMUNITY



# Today's Speakers



Brent Westwood  
Quality Assurance Analyst  
Southern Company

YOUR INDEPENDENT MICRO FOCUS SOFTWARE USER COMMUNITY



# House Keeping

- This “LIVE” session is being recorded  
Recordings are available to all Vivit members
- Session Q&A:  
Please type questions in the Questions Pane

YOUR INDEPENDENT MICRO FOCUS SOFTWARE USER COMMUNITY



# Webinar Control Panel

Toggle View Window  
between Full  
screen/window mode.

Questions



# Building a Reusable Automated Testing Framework for UFT

Brent Westwood  
January 23, 2019





# Agenda

- Southern Company Approach to Automated Testing
- Why We Created Our Automation Framework
- What We Created
- Demonstration
- Questions?





# About Southern Company

Southern Company (NYSE: SO) is nationally recognized as a leading energy company based out of Atlanta, Georgia.

- Electric operating companies in 4 states
- Natural gas distribution companies in 7 states
- Generation company serving wholesale customers across America
- Serving 9 million customers through its subsidiaries
- 46,000 megawatts of generating capacity
- 1,500 billion cubic feet of combined natural gas consumption
- 200,000 miles of electric transmission and distribution lines
- 80,000 miles of natural gas pipeline

Source (1/21/19): <https://www.southerncompany.com/about-us/our-business.html>



---

# Our Approach to Automated Testing

---



# Automated Testing Approach

- **Assess** the automated testing request
- **Determine** if the request seems **suitable** for automation
- **Supply** an automation **framework**
- **Build examples** to prove the automation framework will work with the AUT
- **Train** the application support analyst on how to use the automation framework
- **Enhance** the framework to provide more functionality over time

\*AUT = Application Under Test



# Assessing Suitability for Automation

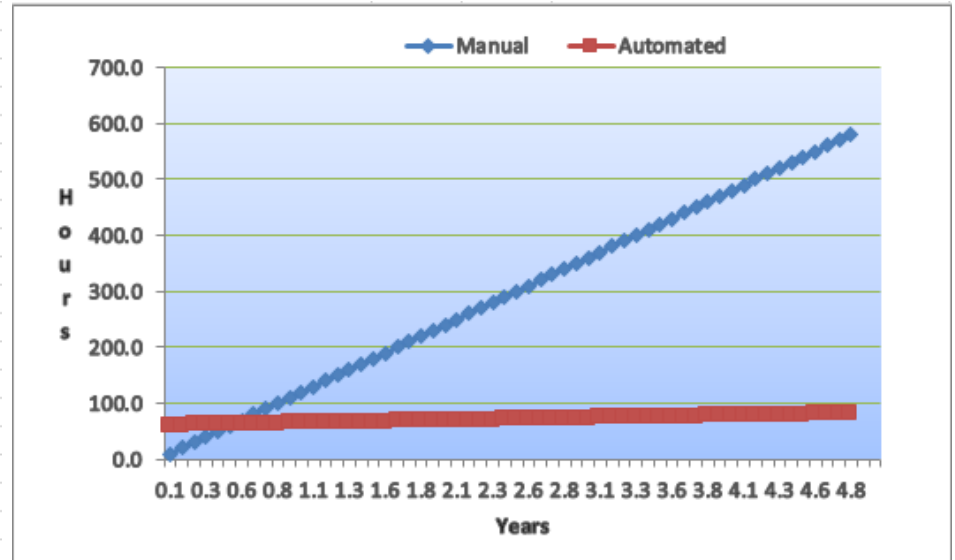
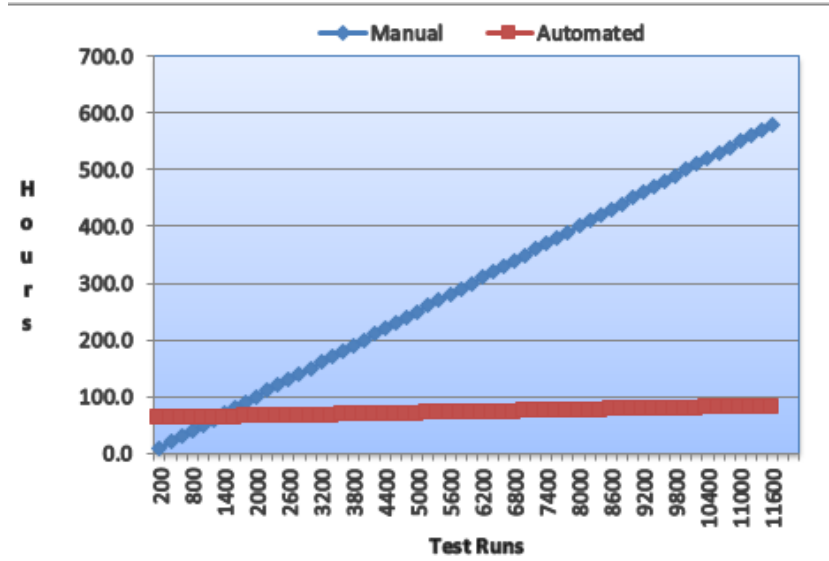
- Tests can be **repeated** – the output can be predicted
- The application(s) involved are **not going to be retired soon**
- The application(s) involved **will not be modified often**
- The application(s) involved are **compatible** with our automated testing tool, Micro Focus UFT
  - Spy the application(s) to determine if UFT can recognize the elements/controls on the screens of the application(s)
- Tests **will be run many times** and/or involve many transactions
  - In order to obtain a **return on investment**, there should be a need to run the automated tests many times or iterations. The more runs, the more time saved (time to run automation vs running the tests manually).



# Return on Investment

Estimate the ROI (Gain from investment – Cost of the investment)

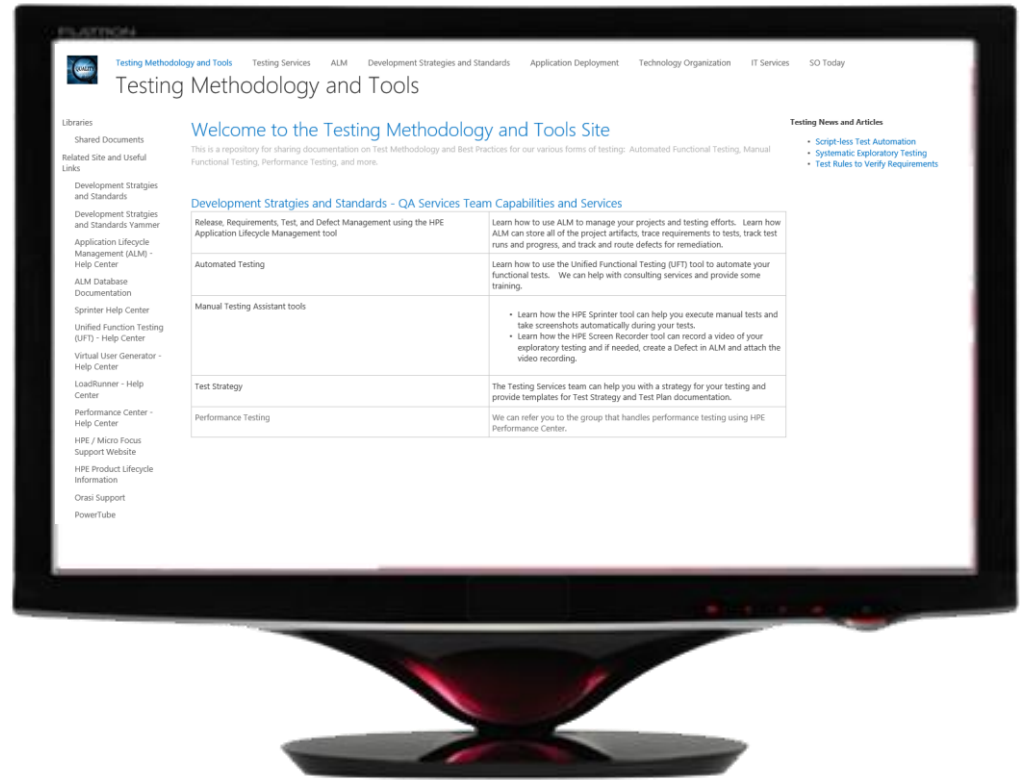
- Graphical representation of the cost to run tests manually vs automated over time is a good way to show the value of automating the tests.



# Training and Help

Have a central location for:

- Help on ALM and Automated Testing (Micro Focus links)
- Automated Testing Knowledgebase
- UFT installation instructions and Chrome setup
- Automation Framework documentation



# Some Keys to Successful Automated Testing

- Working closely with the developer
  - Developer can add HTML ID attributes to the HTML elements which greatly increases object recognition
  - Quick turnaround on updating objects and fixing bugs (blockers)
- Well defined manual tests
  - Know exactly what to do before applying automation
- Access to the database
  - Be able to query the database to get data and verify updates made to records via actions made in the application under test are saved (adds, updates, deletes)
- Test user IDs for the application testing
  - Ability to establish test login IDs for use in testing different security roles
  - Do not have to use your personal network ID and password
  - Allows for easier testing and handoff



---

# Automated Testing Framework

## Why We Created the Framework

---





# Why We Created the Framework

Feature	
Readable Tests	<ul style="list-style-type: none"><li>• Automated tests that are readable and easy to understand</li><li>• The manual tester can create the instructions for the automated test (test steps)</li><li>• Manual tests and automated tests can be the same set of instructions</li></ul>
Flexible Development	<ul style="list-style-type: none"><li>• Easy to reorder the steps and insert new steps as needed</li><li>• Can start at any point in the test and skip steps</li><li>• Easy to create similar tests such as tests that have a different order of operations but use the same fields</li></ul>
Less of a Learning Curve	<ul style="list-style-type: none"><li>• Tester writes test step instructions, not code</li><li>• Tester just needs to know how to identify and define objects (HTML elements)</li><li>• The code to interpret actions is already written</li><li>• The code for the framework is enhanced by those with the ability to code</li></ul>
Code Once	<ul style="list-style-type: none"><li>• Code for interacting with buttons, textboxes, hyperlinks, etc. is coded just one time</li><li>• Code is reusable across all web application projects</li><li>• One set of code to maintain as opposed to many scripts/components</li></ul>



# Why We Created the Framework

Feature	
Logical Object Repository	<ul style="list-style-type: none"><li>• One logical Application Object Repository that can be easily filtered and searched</li></ul>
Object Reuse	<ul style="list-style-type: none"><li>• UFT Repository Objects are added once per Object Class and identifying property</li><li>• The identifying property <u>value</u> of the shared UFT repository object is just changed (<i>.SetTOProperty</i>) per element on the screen to click or enter text</li></ul>
Test Set Parameters	<ul style="list-style-type: none"><li>• Ability to provide parameters at the Test Set level</li><li>• Parameter values can be used by all tests</li><li>• Parameter values and data can be dynamically created by tests</li></ul>
Parameter Reuse	<ul style="list-style-type: none"><li>• Parameter values can be defined in one place as opposed to having to update in every test or component</li><li>• Test parameters are saved and can be used at any time as opposed to only being saved in MEMORY and lost at the end of a test run</li><li>• Parameter data can be changed in the middle of the test (update and run from a point in the test)</li></ul>



# Why We Created the Framework

## Feature

- |                     |  |
|---------------------|--|
| Uniform Test Design | <ul style="list-style-type: none"><li>• All tests for any applications will have the same layout</li><li>• Easy to support any test since they are all created in the same way</li></ul>   |
| Uniform Reporting   | <ul style="list-style-type: none"><li>• All steps executed by a test are explicitly reported in the run result report (reporter statements)</li><li>• Test run result reports are easy to understand, great for auditing, have screen shots, and are the same consistent look for all projects</li><li>• Reporting is as easy to increase or decrease as adding or removing rows in a spreadsheet</li></ul>  |
| Error Handling      | <ul style="list-style-type: none"><li>• Run report lists the Test, Test Step, and values used in the step where the error occurred</li><li>• All identifying properties of the object used in the step are reported as well as a screen capture at the point of the failure</li><li>• Easy to understand exactly what went wrong and where to fix the problem</li><li>• Using the SKIP step ability, one can then update the test or object identification and run the test from the failed step forward</li></ul> |



---

## What We Created

---



# A Readable Automated Test

- The test is a set of readable instructions:
  - **Actions** -- “On Page”, “On Object” - “Perform Action”
  - **Functions** -- such as “Close All Browsers” and “Launch Browser”

A	B	C	D	E	F	G	H
Skip_Step	Step_Label	Call_Function	On_Page	On_Object	Of_Object_Type	Perform_Action	Use_Value
		CloseAllBrowsers					
		LaunchBrowser					[BrowserType][URL]
			Login	Username	Textbox	Enter Text	[LoginUsername]
			Login	Password	Textbox	Enter Password	[LoginUserPassword]
			Login	Login	Button	Click	
			Home	Requests for Approval	Page Header	Verify Exists	
		Take Screenshot					SCREENSHOT - Log In Successful   Logged into CC as [LoginUsername] on machine [MACHINENAME].



# The Framework is Composed of...

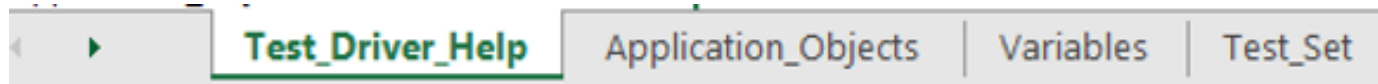
- Help worksheet
- Application Objects worksheet
- Variables worksheet
- Test Set worksheet
- Test worksheets



# Help Worksheet

Defines each section of the framework spreadsheet:

- How to use and setup the ...
  - **Application Objects** worksheet
  - **Variables** worksheet
  - **Test Set** worksheet
  - **Test** worksheets
- Action definitions
- Function definitions
- Object type definitions
- Available template values that are translated at runtime



# Application Object Worksheet

- Defines the objects for any test -- how they are seen to the tester/user
- Application Objects are organized by Page (**Page|Object\_Label|Object\_Type**)
- The Application Objects reference UFT Repository Objects
- The UFT Repository Objects are setup per Object Class (WebEdit, WebList, WebElement, Link, Image).
  - For example, one WebEdit object may be used to handle any Textbox control in the whole website.

Test_Driver_Help	<b>Application_Objects</b>	Variables	Test_Set
------------------	----------------------------	-----------	----------





# Variables Worksheet

- Table to store variable names and values which can be used by tests during a run or during later runs
- Variables can be manually created and updated
  - Such as global object wait, login IDs, database connection info
- Variables can be dynamically created by tests
  - Such as saving values extracted from the screen under test to a new variable for later use
- Variables can be updated by tests
  - Such as a test overwriting the value for preexisting variable
- Variable values exist even after all tests end (not just a value in memory)

Test\_Driver\_Help

Application\_Objects

**Variables**

Test\_Set



# Test Set Worksheet

- Lists the tests to run or skip
- Lists the Test Name, Description, and a column to enter notes or comments
- Lists the Last Run, Result, and any error details for the last run (failed step details)



# Test Worksheet

- Lists all of the steps of the test which can be:
  - an action to take on an object such as
    - sending a click event
    - entering a value in an element on a page
    - extracting a property value from an element on a page
    - comparing a property value to an expected value
  - a function to call such as
    - executing a SQL query, launching a browser, closing browsers, creating a formatted date, etc.
- Any step can be skipped
  - allows one to start a test in the middle of the test
  - allows one to not run steps already developed when adding new steps to the test

Test\_Driver\_Help

Variables

Application\_Objects

Test\_Set

01\_01

01\_02

01\_03

**02\_01**

02\_02

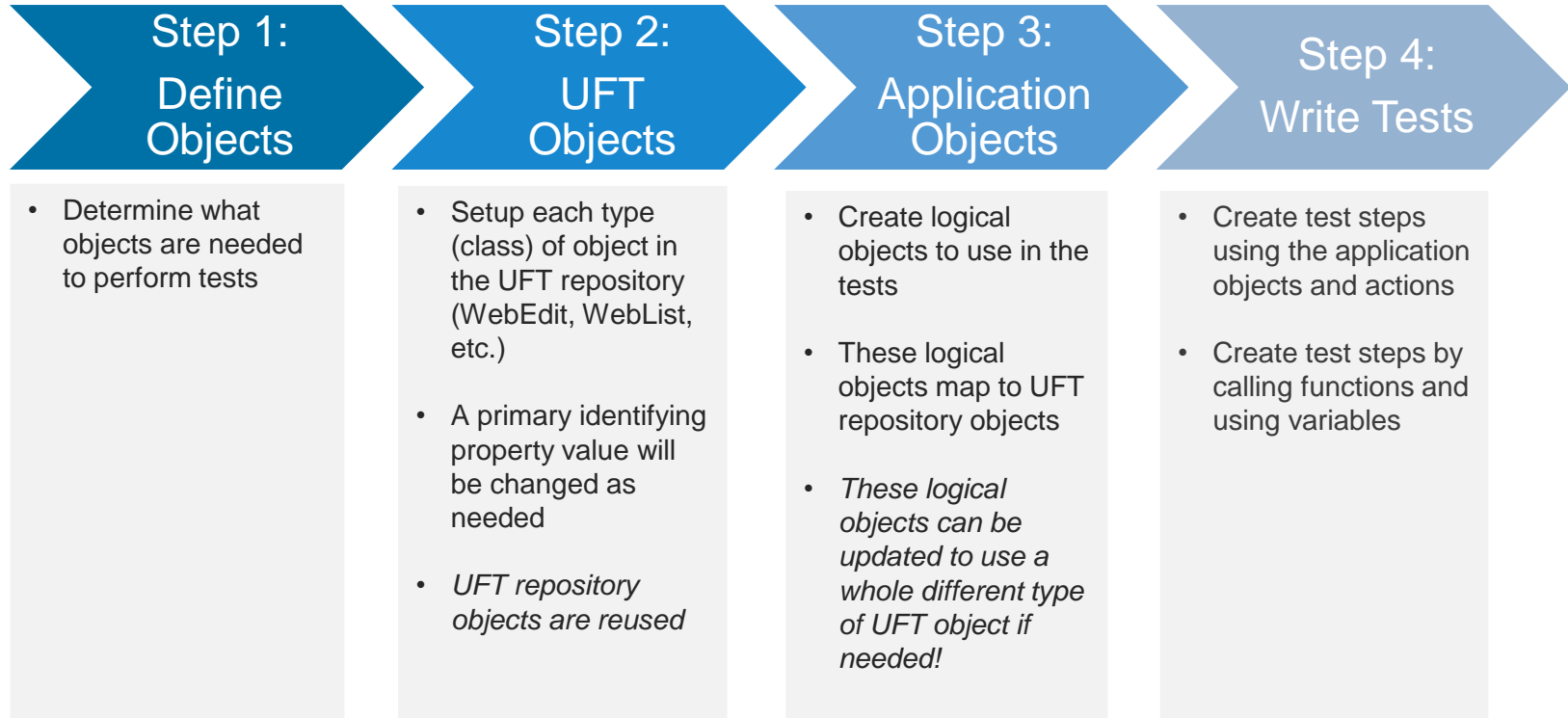
03\_01

03\_02

03\_03

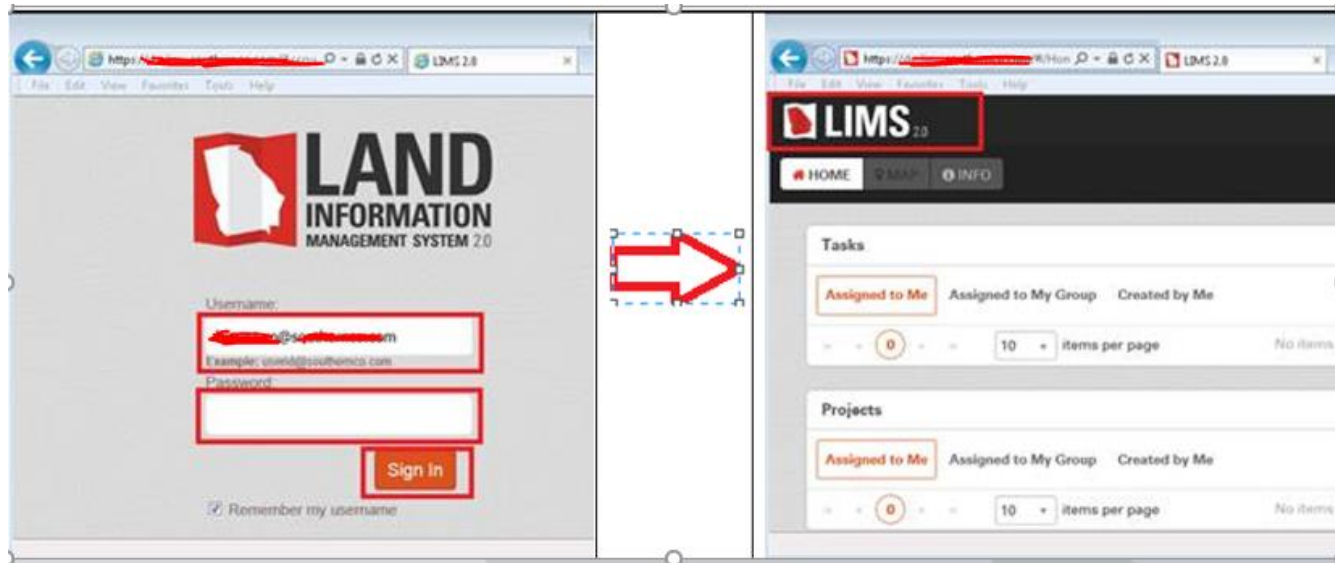


# Building Tests Using the Automation Framework



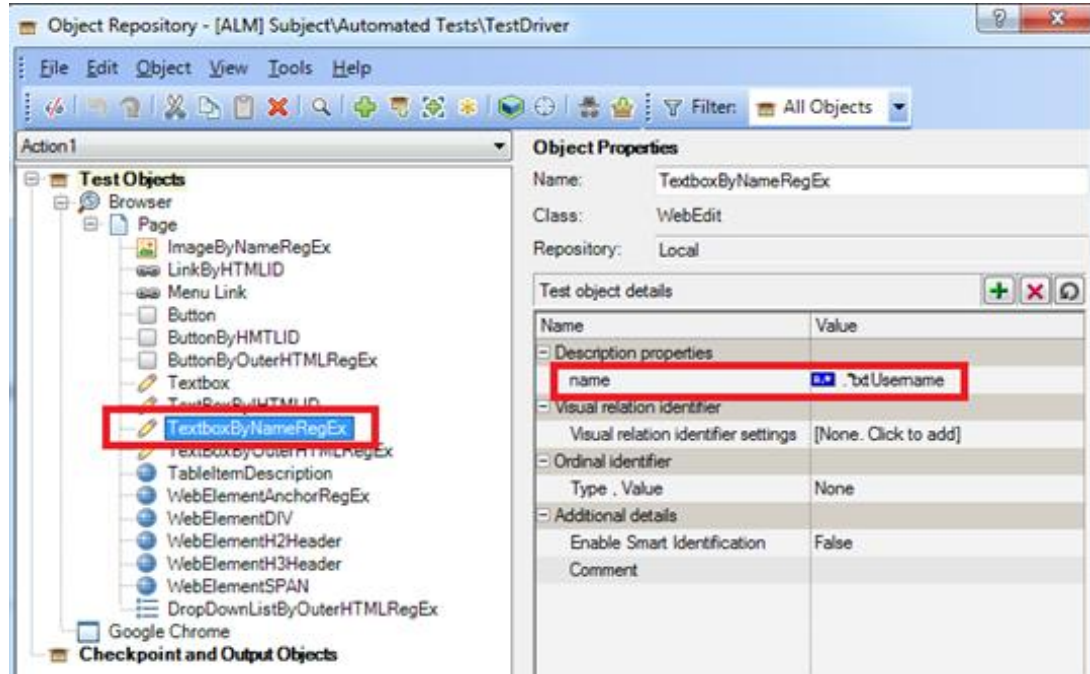
# 1. Determine the Objects Needed

- Example: **Login** test – define the objects needed:
  - Username, Password, and “Sign In” elements on the application’s Login screen
  - Home page image - used to verify that the login in process was successful



## 2. Create Reusable UFT Objects

- Create UFT objects per class of object
- Sample: **WebEdit** object named **TextboxByNameRegEx** that uses the **Name** attribute with a regular expression



### 3. Add Application Object Records

- Add objects that reuse the UFT repository objects
  - the primary property value is replaced with the value from the application object
- Notice the Username and Password objects use the same UFT TextboxByNameRegEx object
- The **Page|Object\_Label|Object\_Type** values are the keys for the **Application\_Object** record and can be referenced in any Test

A	B	C	D	E	F	G	H
Page	Object_Label	Object_Type	Frame	Object_Class	UFT_Object_Name	Primary_Property	Primary_Property_Value
Browser	Browser By Title	Browser		Browser	BrowserByTitle	title	[Parameter_Value]
Login	Username	Textbox		WebEdit	TextboxByNameRegEx	name	.*txtUsername
Login	Password	Textbox		WebEdit	TextboxByNameRegEx	name	.*txtPassword
Login	Sign In	Button		Image	ImageByNameRegEx	name	.*btnLogin

Navigation: Test\_Driver\_Help | **Application\_Objects** | Variables | Test\_Set | Log\_In



## 4. Write Tests

- Call functions like “Close All Browsers” to clean the environment (desktop) and “Launch Browser” to launch a browser to the application website using the browser of choice
- Enter the **Page|Object|Object Type** values with an Action to cause an event on the page element such as entering a value in textbox, clicking a button, clicking a link, getting data out of an element, verifying an element exists, selecting a value from a list, etc.

Skip_Step	Step_Label	Call_Function	On_Page	On_Object	Of_Object_Type	Perform_Action	Use_Value
		LaunchBrowser					[BrowserType][LIMS_URL]
			Login	Username	Textbox	EnterText	[GeneralAccessUserLogin]
			Login	Password	Textbox	EnterPassword	[GeneralUserPassword]
		Take Screenshot					SCREENSHOT - General User Login   Login screen before the General User logs into LIMS.
			Login	Sign In	Button	Click	
			Tasks	Add Task	Button	If Exists Go To Step Label	
			Home	LAND Image	Image Link	Click	

Test\_Driver\_Help

Variables

Application\_Objects

Test\_Set

01\_01

01\_02

01\_03

**02\_01**

02\_02

03\_01

03\_02

03\_03





## 4. Write Tests - Continued

- Create variables on the fly using the **Save\_Value\_To\_Variable** column
- Use the variables later in the same test, in other tests, or in future runs of a test/set of tests

1	Skip_Step	Step_Label	Call_Function	On_Page	On_Object	Of_Object_Type	Perform_Action	Use_Value	Save_Value_To_Variable
11				Add Task	Add Task	Page Header	Verify Exists		
12				Add Task	<b>Task Type</b>	Dropdown List	Select	<b>Master Testing</b>	
13				Add Task	<b>Task Name</b>	Textbox	Enter Text	Auto Test <b>02_01</b> - Task No Default Group [NOW]	<b>TaskName_02_01</b>

1	Skip_Step	Step_Label	Call_Function	On_Page	On_Object	Of_Object_Type	Perform_Action	Use_Value	Save_Value_To_Variable
44				Tasks List	Task Name	Item	Verify Exists	[TaskName_02_01]	
45			Query Application Database For One Result					SELECT J.JOB_ID FROM <del>TABLE</del> WHERE J.JOB_NAME = '[TaskName_02_01]'	<b>TaskID_02_01</b>
46			<b>Take Screenshot</b>					SCREENSHOT - New Task Saved   The new Tasks has been added and verified the record saved to the database. The new Task is listed on the Home page now.	
47				Home	Search	Button	Click		
48				Search	Tasks	Button	Click		
49				Search	Search	Textbox	Enter Text	[TaskID_02_01]	
50				Search	Search Button	Button	Click		
51				Search	Task Edit	Button	Verify Exists	[TaskID_02_01]	



# Test Output – Run Result Report

- Every step is reported
- Filter on the reporter statements to see readable info on every step performed
- Each step maps back to the Test worksheet
- Multiple tests run from the test set are reported in one Run Result Report

The screenshot displays the 'TestDriver - Run Results Viewer' interface. On the left, a list of test steps is shown, with the 46th step, '02\_01 Step 46 - SCREENSHOT - New Task Saved', highlighted in blue. The main area on the right provides a detailed view of this step, indicating it was 'Passed' on 5/17/2018 at 6:19. Below this, a browser window is shown with the URL 'https://[redacted]/#/home#test\_domain\_text'. The browser displays the 'LAND INFORMATION MANAGEMENT SYSTEM 2.0' interface, which includes a search bar, navigation tabs for 'Projects', 'Tasks', and 'My Assigned', and a list of tasks. The task list shows '2018060008 Project Master Testing' with a 'Wk' status, and '688006 Task Master Testing' with a 'Wk' status. The 'Task 68' section is expanded, showing '688006 Task Master Testing Auto Test 02\_01 - Task No Default Group' created on 5/17/2018 at 11:04:54 PM.



# Test Output – Screenshots Only

- Filter by the word “**SCREENSHOT**” to only view reporter statements where screenshots were taken
- Quick way to review and validate the test ran as expected

The image shows a screenshot of a test results viewer application. The left pane displays a tree view of test results for 'TestDriver - LIMS\_Brent 1'. The search filter is set to 'SCREENSHOT', and the results are filtered to show only steps that include this keyword. The selected step is '02\_01 Step 52 - SCREENSHOT - Searched the Task By ID'. The right pane shows the details for this step, indicating it passed. Below the details, there is a browser window showing a search page for 'LIMS 2.0' with the search term '668095'. The browser window also shows a notification for 'HP Functional Testing Agent' debugging the browser.

TestDriver - LIMS\_Brent 1 Test Results (Filtered) - Run Results Viewer

Search for: SCREENSHOT

TestDriver - LIMS\_Brent Summary

- TestDriver - LIMS\_Brent Iteration 1 (Row 1)
- Action1 Summary
- 02\_01 Step 6 - SCREENSHOT - General User Login
- 02\_01 Step 14 - SCREENSHOT - Add Task InitialScreen
- 02\_01 Step 17 - SCREENSHOT - Add Task - Second Screen
- 02\_01 Step 22 - SCREENSHOT - After Clicking Save - Before Entering R
- 02\_01 Step 24 - SCREENSHOT - After Clicking Save - Before Entering R
- 02\_01 Step 30 - SCREENSHOT - Entered Group - Clicked Save
- 02\_01 Step 42 - SCREENSHOT - Entered a Test Domain - Desc - Categ
- 02\_01 Step 46 - SCREENSHOT - New Task Saved
- 02\_01 Step 52 - SCREENSHOT - Searched the Task By ID

Result Details

Step Name: 02\_01 Step 52 - SCREENSHOT - Searched the Task By ID

Step Passed

Object	Details	Result	Time
02_01 Step 52 - SCREENSHOT - Searched the Task By ID	The new Task was saved to the database and found by searching for the Task in LIMS by Task ID: Task1 D: 668095	Passed	5/17/2018 6:32

Result Details | Screen Recorder | System Monitor

General View

LIMS 2.0

Secure | [Redacted] /Home#test\_domain\_text

HP Functional Testing Agent is debugging this browser. Cancel

Search Tasks Projects 668095

Sort

Created Da • BY Most Recen

Filter

Status

Task Type

Assignee

Assigned Group

Auto Test 02\_01 - Task No Default Group 5/17/2018 11:04:4

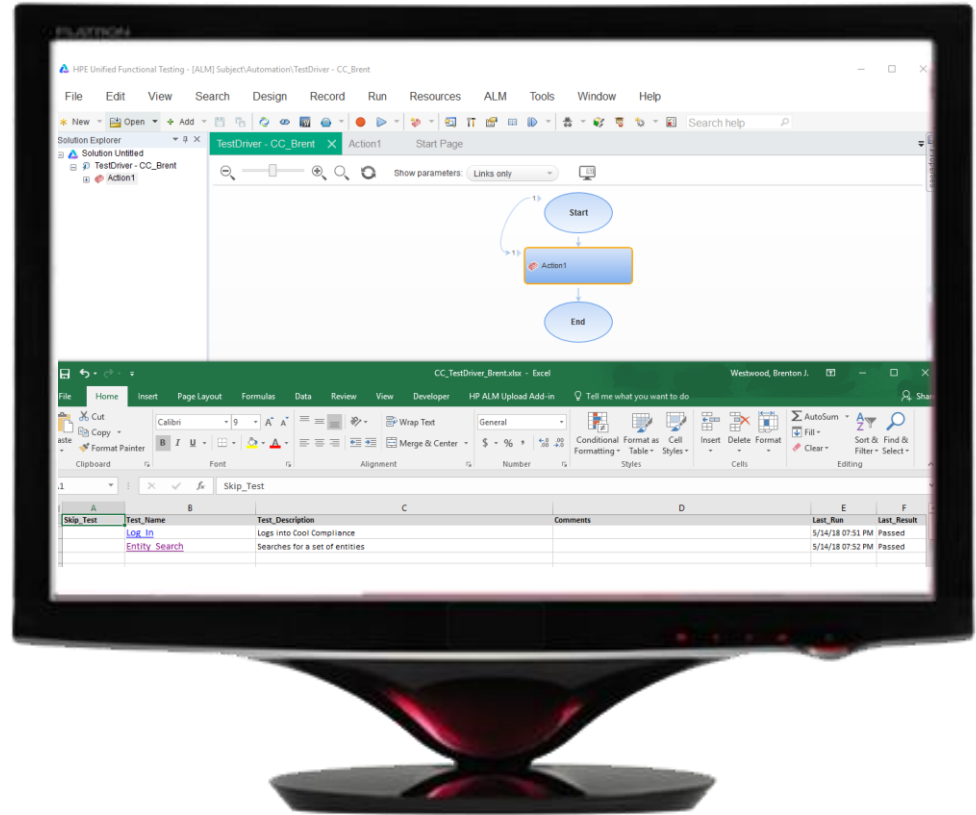
668095 TASK MASTER TESTING DUE N/A ASSIGNED PATRICK #24 PENT LAND COM

Add Task Test Without Default Group by bjwestwo at 5/17/2018 11:05:30 PM

1 8 items per page

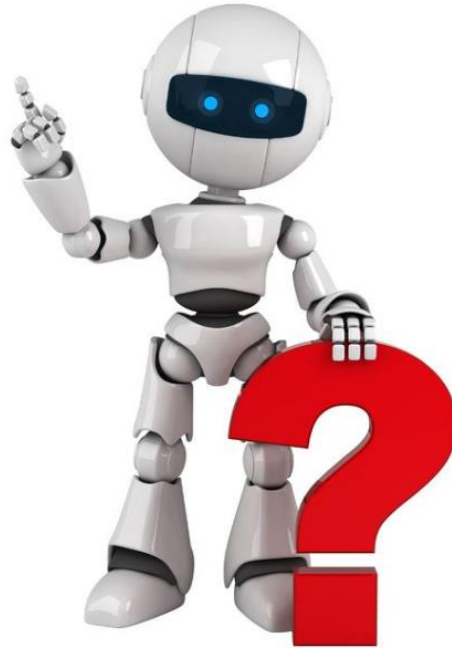


# Demonstration



# Open Discussion

Any Questions?



Thank You,

Brenton Westwood

Reach me on Linked In: <https://www.linkedin.com/in/brenton-westwood-b3823b54/>



# ITOM Summit 2019

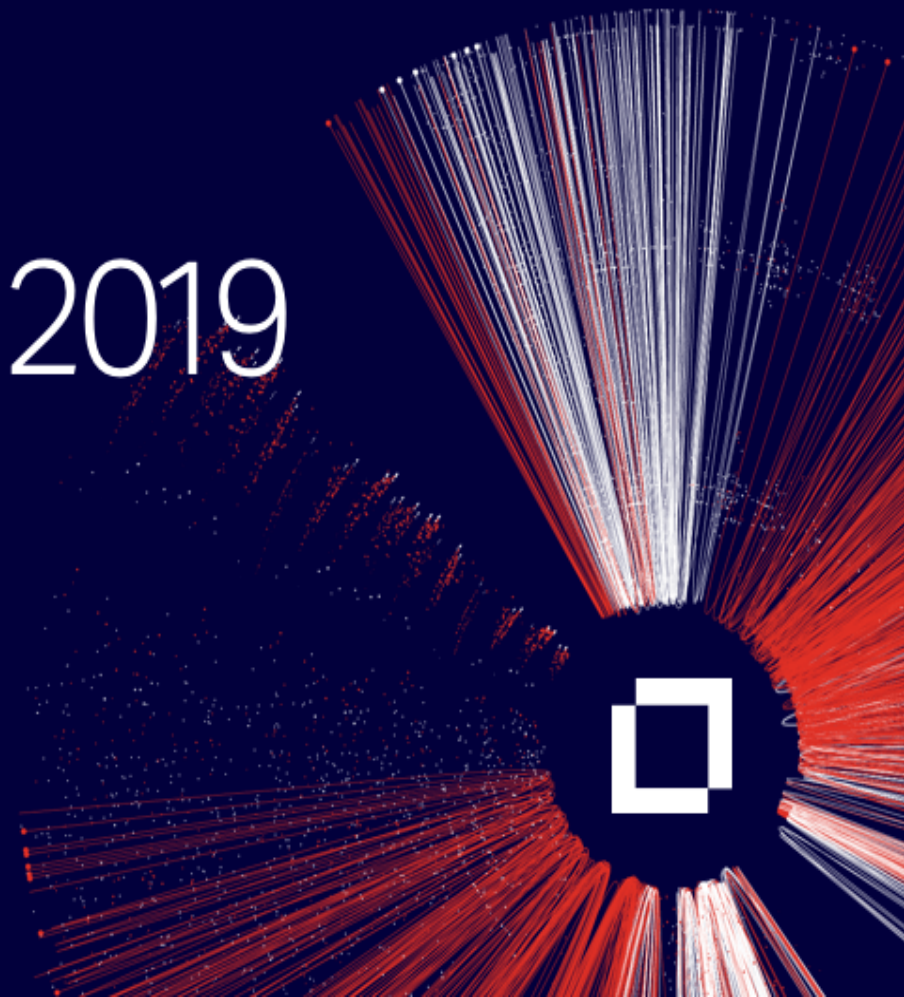
February 5–7

Pointe Hilton Tapatio Cliffs Resort, Phoenix, AZ

Register today

<http://bit.ly/2TRufdG>

#ITOMsummit



# Upcoming Vivit Events

January 24, 2019

**Meeting: Discover the New and Exciting Micro Focus - Florida LUG Event**

5:30 - 7:00 PM EST (Florida)

<https://www.vivit-worldwide.org/events/EventDetails.aspx?id=1172751&group=>

January 31, 2019

**Meeting: User Group Treffen "Testen & Test-Management mit Produkten von Micro Focus in München**

13:00 - 17:00 CET

<https://www.vivit-worldwide.org/events/EventDetails.aspx?id=1180268&group=>

YOUR INDEPENDENT MICRO FOCUS SOFTWARE USER COMMUNITY





# Thank you

- Complete the short survey so your Vivit leaders can better serve you in the future

[www.vivit-worldwide.org](http://www.vivit-worldwide.org)

YOUR INDEPENDENT MICRO FOCUS SOFTWARE USER COMMUNITY





**Thank You**  
vivit-worldwide.org

