



Quick Start Guide

for Windows

Examples with Excel files

Some Useful Definitions

- **AUT:** Application Under Test
- **AUT Launch File:** the configuration file used by the Maveryx Test Automation Framework to run the related AUT
- **Keyword-driven testing:** a codeless approach to write test cases for non-programmers
- **Data-driven testing:** a methodological approach to separate test cases from test data

Requirements

To work with Maveryx, your system shall meet the following minimum requirements:

- Windows 7 or later
- Java Runtime Environment ver. 1.8.0_211
(<https://www.java.com/en/download/>)
- .NET Runtime Environment ver. 4.6 or later
(<https://www.microsoft.com/en-us/download/details.aspx?id=48130>)

Summary

- Install and configure Maveryx
- Get license key
- Create and run your first test

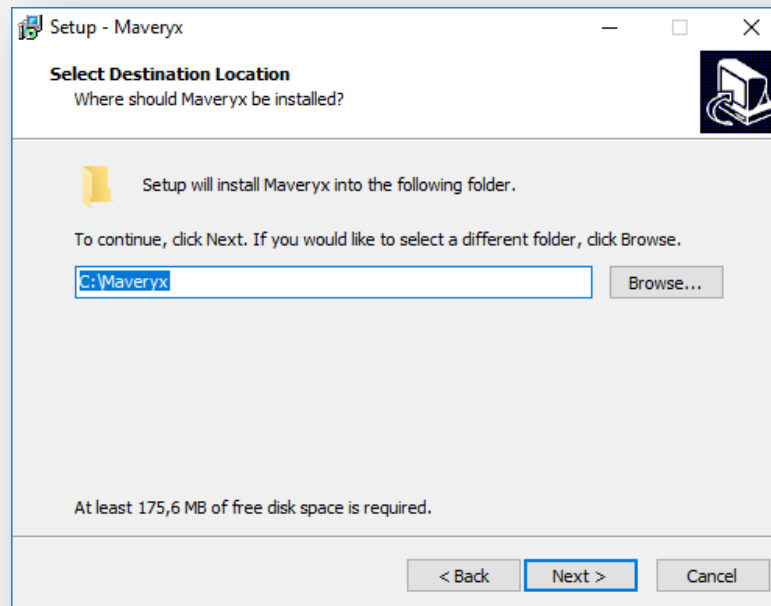
Cap I

- **Install and configure Maveryx**
- Get license key
- Create and run your first test

Installation (1)

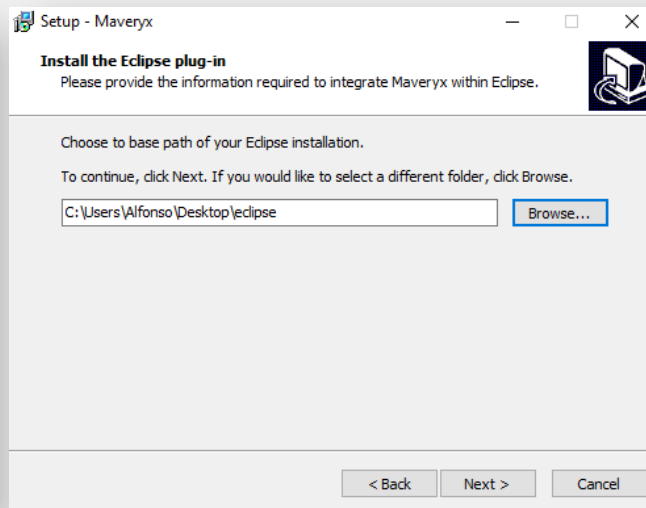
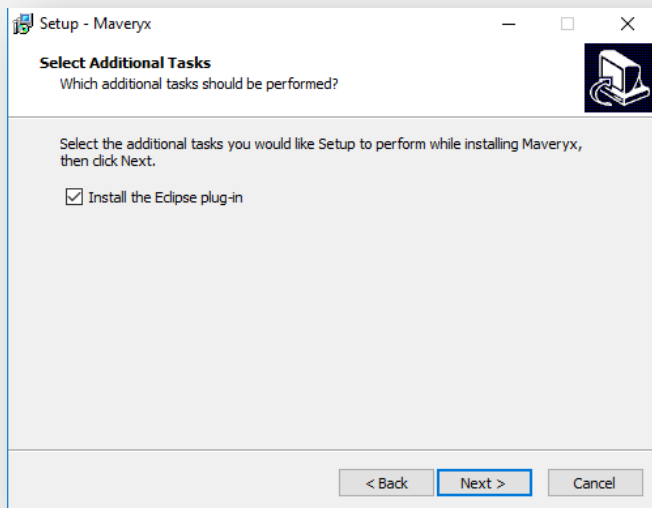
To install the Maveryx Test Automation Framework on your system, run **Maveryx_Win_2.X.y_Trial.exe** and follow the steps of the setup wizard.

Choose the directory into which you want to install the Maveryx software. You must have write permissions to this directory.



Installation (2)

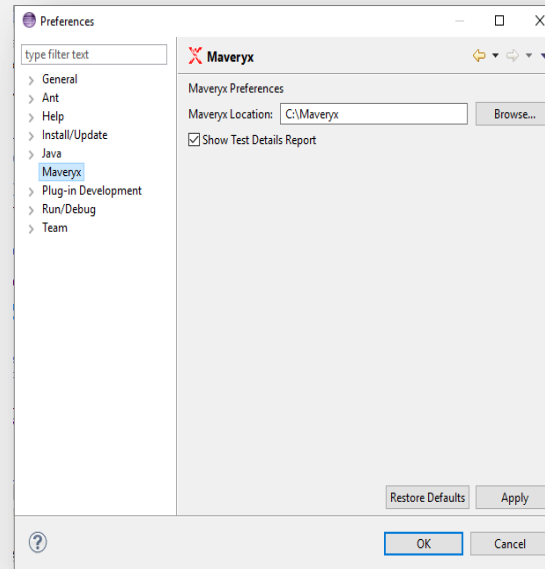
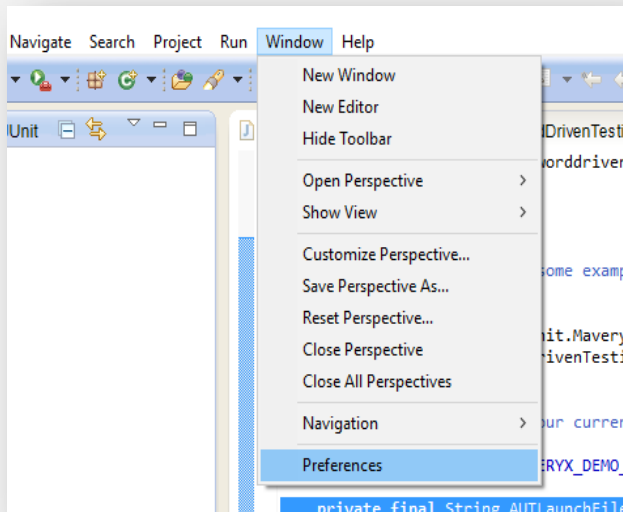
You can skip this step but, in case, you can also **Install the Eclipse plug-in** by clicking on the checkbox and selection the Eclipse path.



If you want to install the Maveryx Eclipse Plugin later, copy the files in **MAVERYX_HOME\tools\EclipsePlugin** folder into the **/dropins** directory of your Eclipse installation.

Setting Up Eclipse

- Make sure that the Maveryx Eclipse Plugin files
 - com.maveryx.ide_2.0.1.202004010017.jar
 - com.maveryx.report.chart.lib_2.0.1.202004010017.jar
 - com.maveryx.report.lib_2.0.1.202004010017.jarare in the **/dropins** folder of your Eclipse installation directory
- Run Eclipse and
 - Click on "**Window > Preferences**" menu on the menu bar to open the Preferences dialog
 - Select the item "**Maveryx**" to open the Maveryx's preferences page
 - Click "**Browse...**" to select the Maveryx installation directory
 - Click "**OK**"



Cap II

- Install and configure Maveryx
- **Get license key**
- Create and run your first test

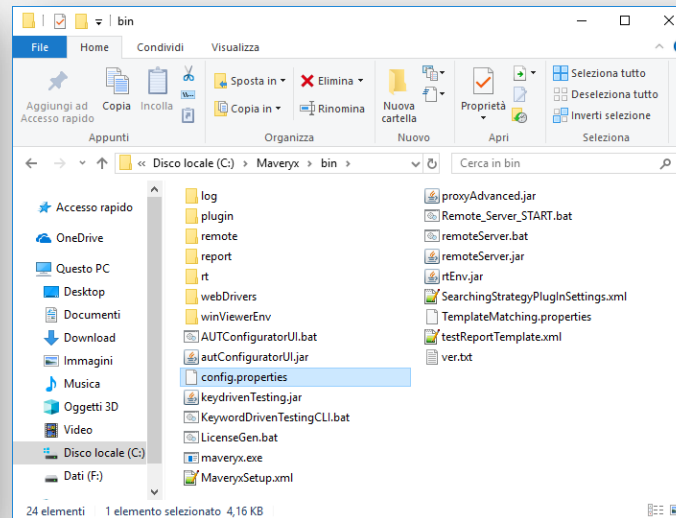
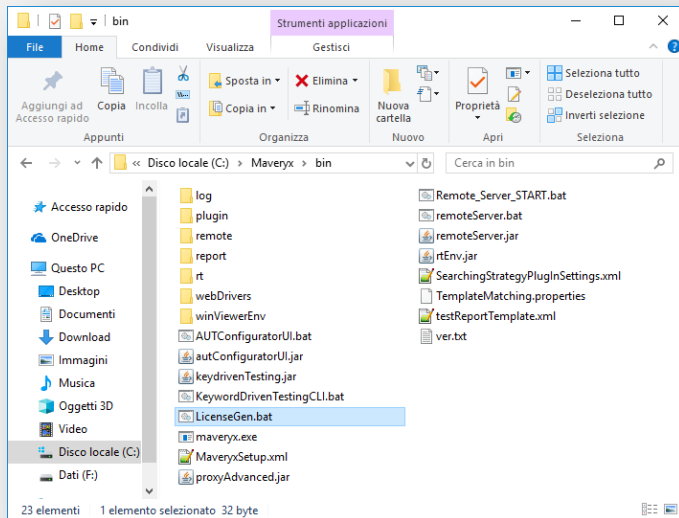
Maveryx License

- Maveryx uses a **Node-Locked** license type. A node-locked license for Maveryx lets you run the application on a specific machine or workstation. This license type is considered a single-user license, although it's bound to the machine, not the user.
- Trial versions of Maveryx (which is always licensed as Node-Locked) have a time-limited license. After it expires, you can no longer use the product.
- After a commercial license for Maveryx expires, you can continue using the product. However, you will not be able to get updates for the product and technical assistance from the Maveryx Support team.

Collecting Node Data

To generate a license open **MAVERYX_HOME/bin/** folder and run the **LicenseGen.bat** file.

This utility will automatically collect all hardware and software information needed to generate a valid Maveryx license by saving them into the **config.properties** file in **MAVERYX_HOME/bin/** folder.

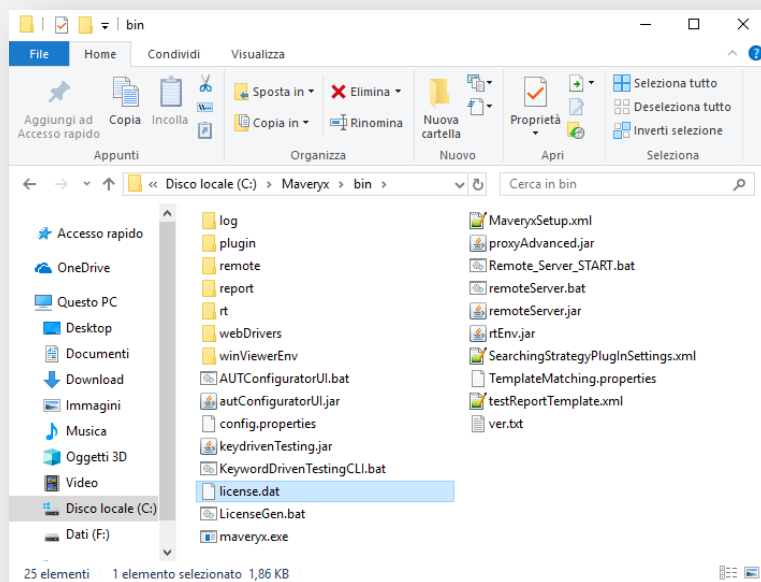


License Key

When you will have the **config.properties** file, send it by email to license.manager@maveryx.com with the subject "**License Key Request**".

In reply to your email, you will receive your license key file (**license.dat**) as attachment.

Save the **license.dat** file into the **MAVERYX_HOME/bin/** folder.



Cap III

- Install and configure Maveryx
- Get license key
- **Create and run your first test**

Java AUT Launch File

To execute a Java Application-Under-Test it is necessary to create the related **AUT launch file**.

```
<?xml version="1.0" encoding="UTF-8"?>
<AUT_DATA>
  <SERVER_URL></SERVER_URL>

  <WORKING_DIR>./src/resources/AUT/java</WORKING_DIR> <!-- change this path to your working directory -->

  <APPLICATION_NAME>ButtonDemo</APPLICATION_NAME>

  <AUT_ARGUMENTS></AUT_ARGUMENTS>

  <VM_ARGUMENTS></VM_ARGUMENTS>

  <DESCRIPTION>
    Push-Button testing
  </DESCRIPTION>

  <JRE_PATH>${java.home}</JRE_PATH> <!-- change this path to your JRE home -->

  <MAIN_CLASS>com.sun.demo.ButtonDemo</MAIN_CLASS>

  <!-- on UNIX-like and MAC OS X systems change the path separator ';' to ':' -->
  <CLASSPATH>
    <LIB>
      <PATH>examples.jar</PATH> <!-- change this path to your Maveryx installation directory /demo -->
    </LIB>
    <!-- do not change the data below! (except for path separator on UNIX-like and MAC OS X systems) -->
  </CLASSPATH>
</AUT_DATA>
```

MFC & .Net AUT Launch File

To execute a MFC or .NET Application-Under-Test it is necessary to create the related **AUT launch file**.

```
<?xml version="1.0" encoding="UTF-8"?>
<AUT_DATA>
  <EXECUTABLE_PATH>.\src\resources\AUT\windows\notepad Enhanced.exe</EXECUTABLE_PATH>
  <APPLICATION_NAME>Notepad Enhanced</APPLICATION_NAME>
  <TOOLKIT>WIN</TOOLKIT>
  <TIMEOUT>1000</TIMEOUT>
  <DELTA_CHECK>1000</DELTA_CHECK>
  <AUT_ARGUMENTS></AUT_ARGUMENTS>
</AUT_DATA>
```

Set the absolute or relative path to your AUT executable file

Web AUT Launch File

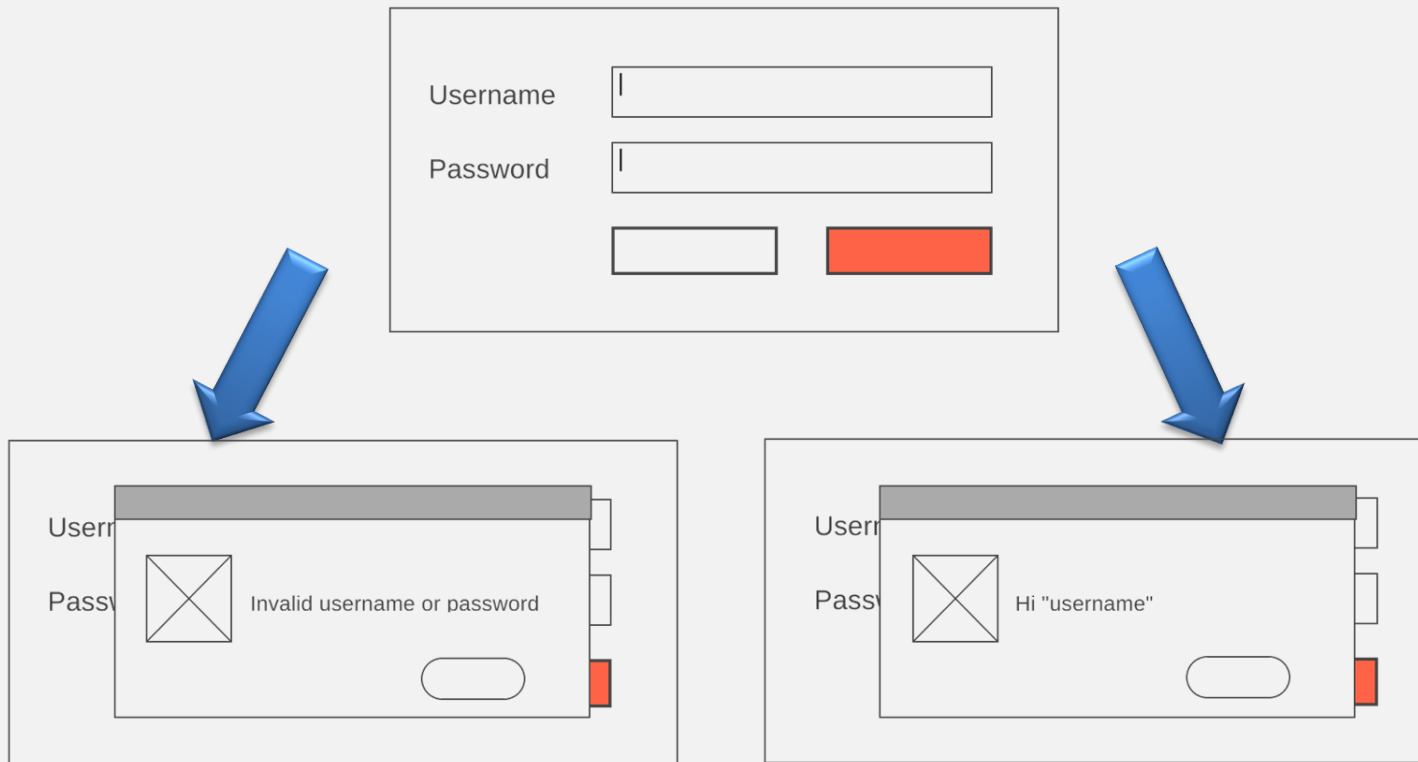
To execute a Web Application-Under-Test it is necessary to create the related **AUT launch file**.

```
<?xml version="1.0" encoding="UTF-8"?>
<AUT_DATA>
  <EXECUTABLE_PATH>C:/Program Files (x86)/Google/Chrome/Application/chrome.exe</EXECUTABLE_PATH>
  <APPLICATION_NAME>CHROME</APPLICATION_NAME>
  <TOOLKIT>WEB</TOOLKIT>
  <AUT_ARGUMENTS>file:///./src/resources/AUT/web/index.html</AUT_ARGUMENTS>
</AUT_DATA>
```

Set the URL of the AUT

Set the path of the browser **you** want to use for your tests

The Sample AUT



Test Case #001

Test Case : TC_01

1. Start the Application
2. Enter valid username
3. Enter valid password
4. Click "Login" button
5. Check the results: Hi "username"
6. Click "OK" button
7. Close the Application



Diagram of the login form. It contains two input fields labeled 'Username' and 'Password', a 'Login' button, and an 'OK' button. Red boxes highlight the 'Username' and 'Password' input fields. Red arrows point from step 2 to the 'Username' field and from step 3 to the 'Password' field.

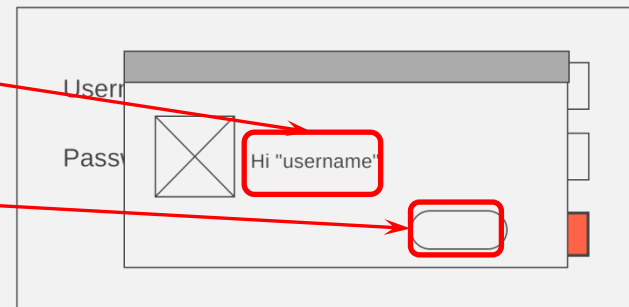


Diagram of the result window. It contains a message box with a close button (X) and a text area displaying 'Hi "username"'. A red box highlights the text area. Red arrows point from step 5 to the text area and from step 6 to the close button.

Identify Keywords

Test Case

1. Start the AUT
2. Enter username
3. Enter Password
4. Click "Login" button
5. Check the results
6. Click "OK" button
7. Close the AUT

Keywords

Start

Set_Text

Click

Has_Text

Close

Design Test step 1

Test Case : TC_01

1. Start the Application

2. Enter valid username
3. Enter valid password
4. Click "Login" button
5. Check the results: "Hi alfonso"
6. Click "OK" button
7. Close the Application

Click on login button with enter valid username and password

	A	B	C	D	E	F	G
1	Test Case ID :	TC_01	Author(s) :	Maveryx	Test Case		
2	Description :	Click on login button w	Requirement(s) :	REQ_1			
3	OBJECT	NAME	CONTAINER	CONTAINER NAME	ACTION	DATA	DATA
4					START	Login	
5							

Keyword	Data / Input	Description
START	AUT lauch file path	Launch the AUT

Design Test step 2 & 3

Test Case : TC_01

1. Start the Application
2. Enter valid username
3. Enter valid password
4. Click "Login" button
5. Check the results: "Hi alfonso"
6. Click "OK" button
7. Close the Application

A diagram of a login form. It contains two text input fields labeled 'Username' and 'Password'. Below these fields are two buttons: a white one on the left and a red one on the right. Red arrows point from the 'SET_TEXT' entries in the table below to the 'Username' and 'Password' input fields.

1	Test Case ID :		Author(s) :	Test Case		
2	Description :		Requirement(s) :			
3	OBJECT	NAME	CONTAINER	CONTAINER NAME	ACTION	DATA
4					START	Login
5	TEXT	Username			SET_TEXT	alfonso
6	PASSWORD_TEXT	Password			SET_TEXT	alfonsopwd
7						

Keyword	Data / Input	Description
SET_TEXT	Text	Set the text into a text field

Design Test step 4

Test Case : TC_01

1. Start the Application
2. Enter valid username
3. Enter valid password
4. Click "Login" button
5. Check the results: "Hi alfonso"
6. Click "OK" button
7. Close the Application

A screenshot of a login form. It contains two input fields: 'Username' and 'Password'. Below the 'Password' field is a 'Login' button, which is highlighted with a red rectangle. A red arrow points from the 'Login' button in the table below to this button in the form.

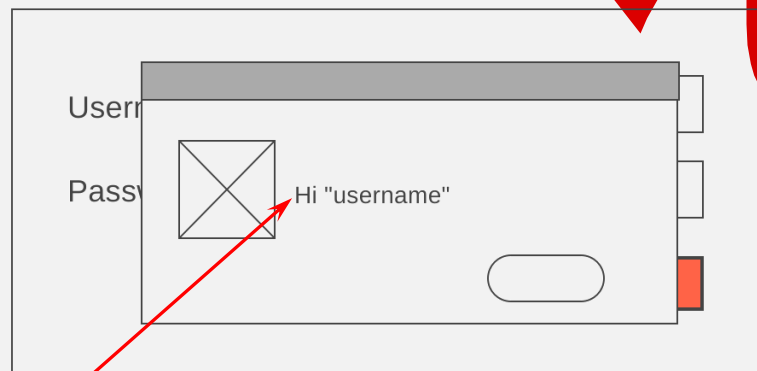
1	Test Case ID :	Author(s) :	Test		
2	Description :	Requirement(s) :			
3	OBJECT	NAME	CONTAINER	CONTAINER NAME	ACTION
4					START
5	TEXT	Username			SET_TEXT
6	PASSWORD_TEXT	Password			SET_TEXT
7	BUTTON	Login			CLICK
8					

Keyword	Data / Input	Description
CLICK		Click the button

Design Test step 5

Test Case : TC_01

1. Start the Application
2. Enter valid username
3. Enter valid password
4. Click "Login" button
5. Check the results: "Hi alfonso"
6. Click "OK" button
7. Close the Application



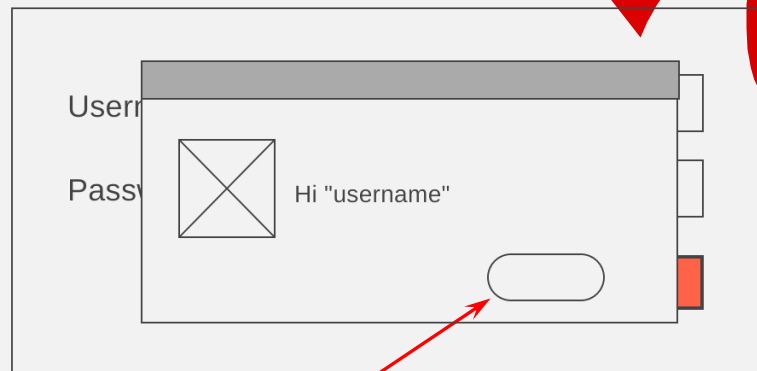
				Test Case		
1	Test Case ID :		Author(s) :			
2	Description :		Requirement(s) :			
3	OBJECT	NAME	CONTAINER	CONTAINER NAME	ACTION	DATA
4					START	Login
5	TEXT	Username			SET_TEXT	alfonso
6	PASSWORD_TEXT	Password			SET_TEXT	alfonsopwd
7	BUTTON	Login			CLICK	
8	LABEL				HAS_TEXT	Hi alfonso
9						

Keyword	Data / Input	Description
HAS_TEXT	Text	Check whether the given LABEL has the actual Text

Design Test step 6

Test Case : TC_01

1. Start the Application
2. Enter valid username
3. Enter valid password
4. Click "Login" button
5. Check the results:
"logged in"
- 6. Click "OK" button**
7. Close the Application



1	Test Case ID :		Author(s) :		Test Case		
2	Description :		Requirement(s) :				
3	OBJECT	NAME	CONTAINER	CONTAINER NAME	ACTION	DATA	DATA
4					START	Login	
5	TEXT	Username			SET_TEXT	alfonso	
6	PASSWORD_TEXT	Password			SET_TEXT	alfonsopwd	
7	BUTTON	Login			CLICK		
8	LABEL				HAS_TEXT	Hi alfonso	
9	BUTTON	OK			CLICK		
10							

Keyword	Data / Input	Description
CLICK		Click the button

Design Test step 7

Test Case : TC_01

1. Start the Application
2. Enter valid username
3. Enter valid password
4. Click "Login" button
5. Check the results:
"logged in"
6. Click "OK" button
7. Close the Application

1	Test Case ID :		Author(s) :		Test Case		
2	Description :		Requirement(s) :				
3	OBJECT	NAME	CONTAINER	CONTAINER NAME	ACTION	DATA	DATA
4					START	Login	
5	TEXT	Username			SET_TEXT	alfonso	
6	PASSWORD_TEXT	Password			SET_TEXT	alfonsopwd	
7	BUTTON	Login			CLICK		
8	LABEL				HAS_TEXT	Hi alfonso	
9	BUTTON	OK			CLICK		
10					CLOSE	Login	
11							

Keyword	Data / Input	Description
CLOSE	ApplicationName	Close the AUT

Run the Tests

By command line ***KeywordDrivenTestingCLI*** with the following arguments:

For example:

KeywordDrivenTestingCLI "C:\Test\test001.xls"

The test script *C:\Test\test001.xls* is executed

Or:

KeywordDrivenTestingCLI -e "C:\Test\test001.xls" "C:\Test"

All test scripts in *C:\Test* (including subfolders) are executed except *C:\Test\test001.xls*

THANK YOU

sales@maveryx.com

info@maveryx.com

+39 333 30 72 597

+39 351 87 85 706

