MAVERYX®

Test it simple!

Quick Start Guide

for Windows

Examples with MBlockly

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
- **Low-Code testing**: a scriptless approach to write test cases based on blocks programming

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 your first test with MBlockly

Cap I

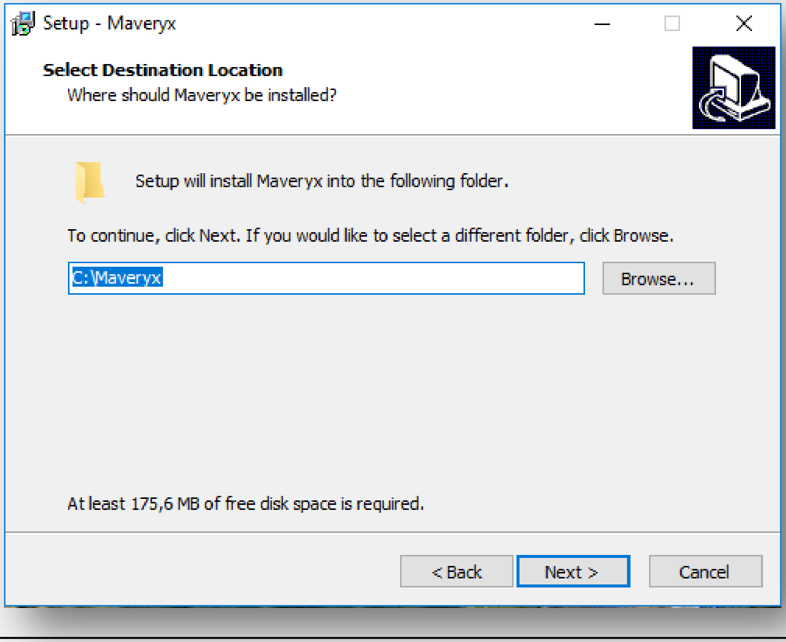
- **Install and configure Maveryx**
- Get license key
- Create your first test with MBlockly

Test it simple!

Installation (1)

To install the Maveryx Test Automation Framework and its Eclipse Plugin 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.



Setup - Maveryx

Select Destination Location

Where should Maveryx be installed?

Setup will install Maveryx into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

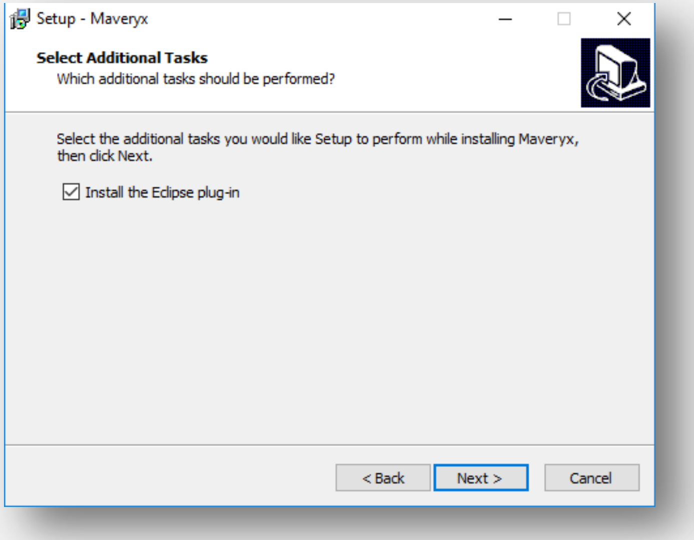
C:\Maveryx

At least 175,6 MB of free disk space is required.

Test it simple!

Installation (2)

You may click on **Install the Eclipse plug-in** to install the Maveryx Eclipse Plugin, then select the Eclipse installation directory.

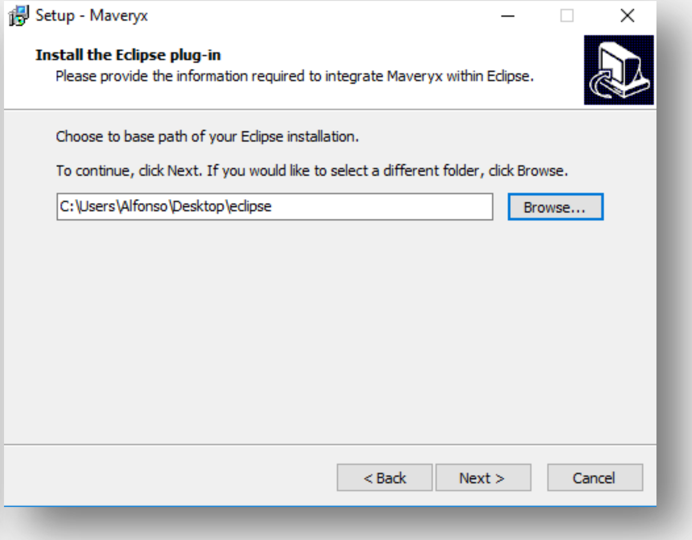


Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Maveryx, then click Next.

☑ Install the Eclipse plug-in



Install the Eclipse plug-in

Please provide the information required to integrate Maveryx within Eclipse.

Choose to base path of your Eclipse installation.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Users\Alfonso\Desktop\eclipse

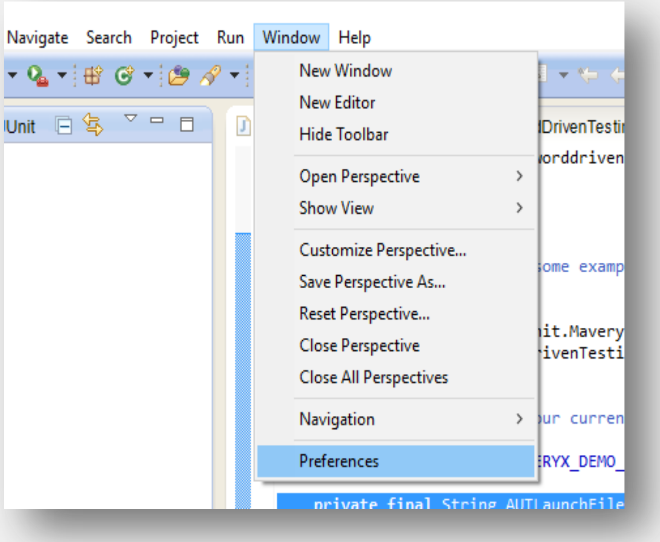
If you want to install the Maveryx Eclipse Plugin later, copy the files in ***MAVERYX_HOME\tools\EclipsePlugin*** folder into the **/dropins** directory of your Eclipse installation.

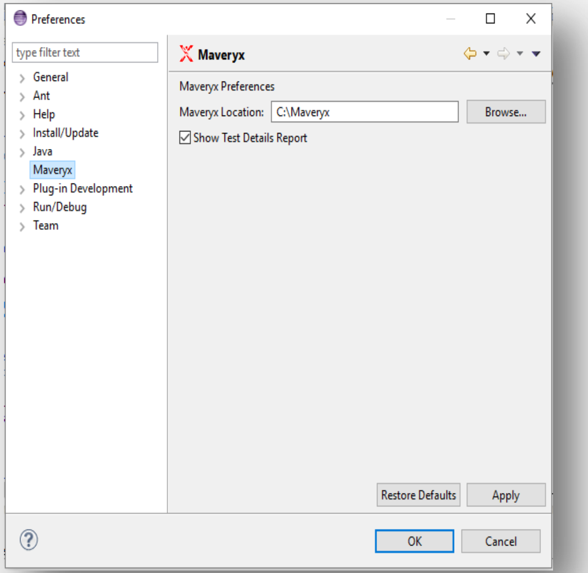
Setting Up Eclipse

- 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.jar

  are 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**"





Maveryx Preferences

Maveryx Location: C:\Maveryx

☑ Show Test Details Report

Cap II

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

Test it simple!

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.

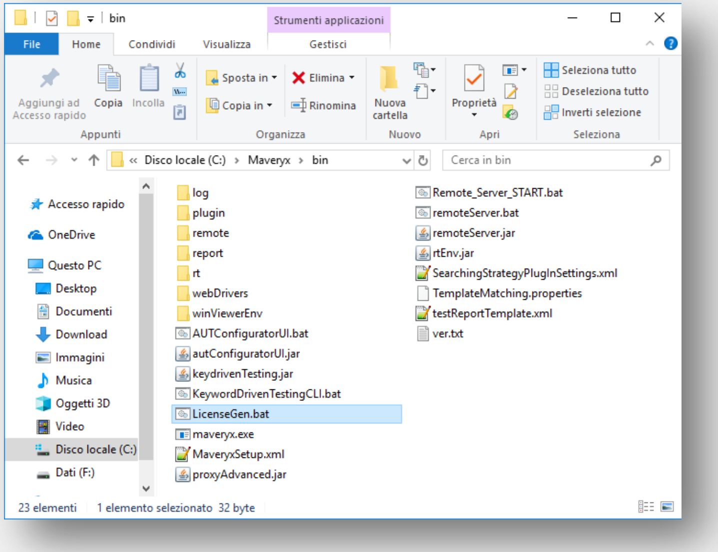
- Trial versions of Maveryx (which is always licensed as Node-Locked) have a time-limited license. After it expires, you can no longer use the product.

- 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

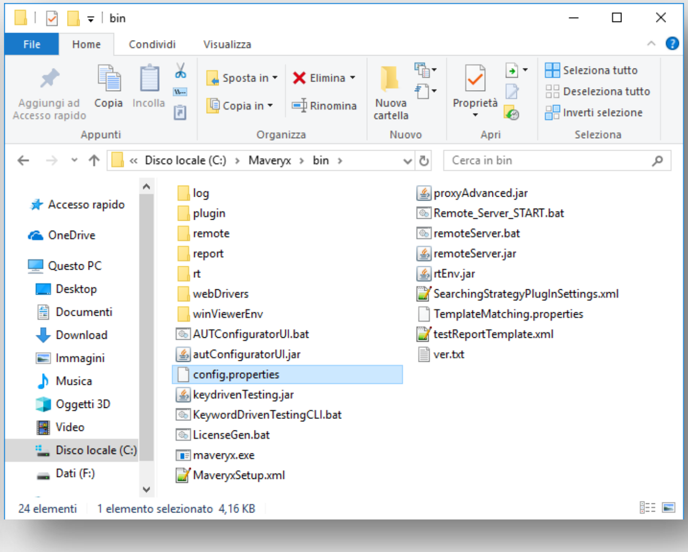
To generate a license open **MAVERYX_HOME/bin/** folder and run the **LicenseGen.bat** file.

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.



log
plugin
remote
report
rt
webDrivers
winViewerEnv
AUTConfiguratorUI.bat
autConfiguratorUI.jar
keydrivenTesting.jar
KeywordDrivenTestingCLI.bat
LicenseGen.bat
maveryx.exe
MaveryxSetup.xml
proxyAdvanced.jar
Remote_Server_START.bat
remoteServer.bat
remoteServer.jar
rtEnv.jar
SearchingStrategyPlugInSettings.xml
TemplateMatching.properties
testReportTemplate.xml
ver.txt

23 elementi 1 elemento selezionato 32 byte



log
plugin
remote
report
rt
webDrivers
winViewerEnv
AUTConfiguratorUI.bat
autConfiguratorUI.jar
config.properties
keydrivenTesting.jar
KeywordDrivenTestingCLI.bat
LicenseGen.bat
maveryx.exe
MaveryxSetup.xml
proxyAdvanced.jar
Remote_Server_START.bat
remoteServer.bat
remoteServer.jar
rtEnv.jar
SearchingStrategyPlugInSettings.xml
TemplateMatching.properties
testReportTemplate.xml
ver.txt

24 elementi 1 elemento selezionato 4,16 KB

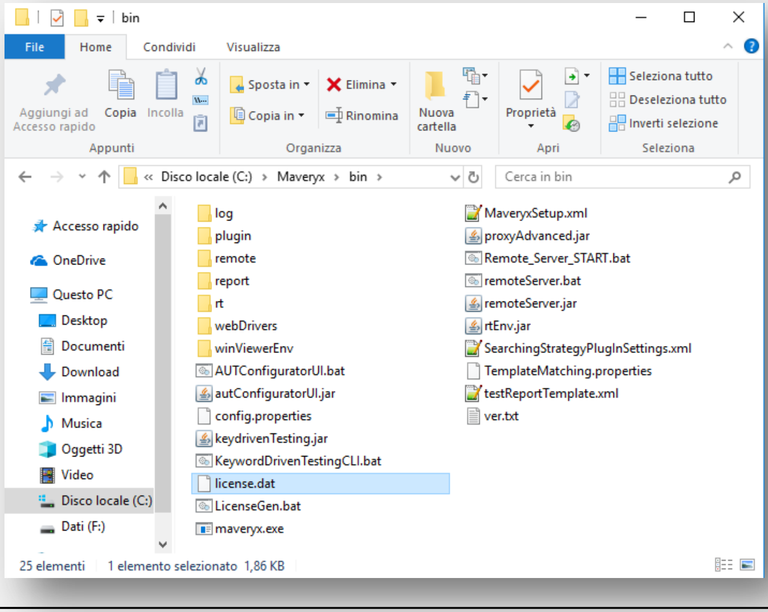
Test it simple!

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.



« Disco locale (C:) › Maveryx › bin ›

log
plugin
remote
report
rt
webDrivers
winViewerEnv
AUTConfiguratorUI.bat
autConfiguratorUI.jar
config.properties
keydrivenTesting.jar
KeywordDrivenTestingCLI.bat
license.dat
LicenseGen.bat
maveryx.exe
MaveryxSetup.xml
proxyAdvanced.jar
Remote_Server_START.bat
remoteServer.bat
remoteServer.jar
rtEnv.jar
SearchingStrategyPlugInSettings.xml
TemplateMatching.properties
testReportTemplate.xml
ver.txt

25 elementi | 1 elemento selezionato 1,86 KB

Test it simple!

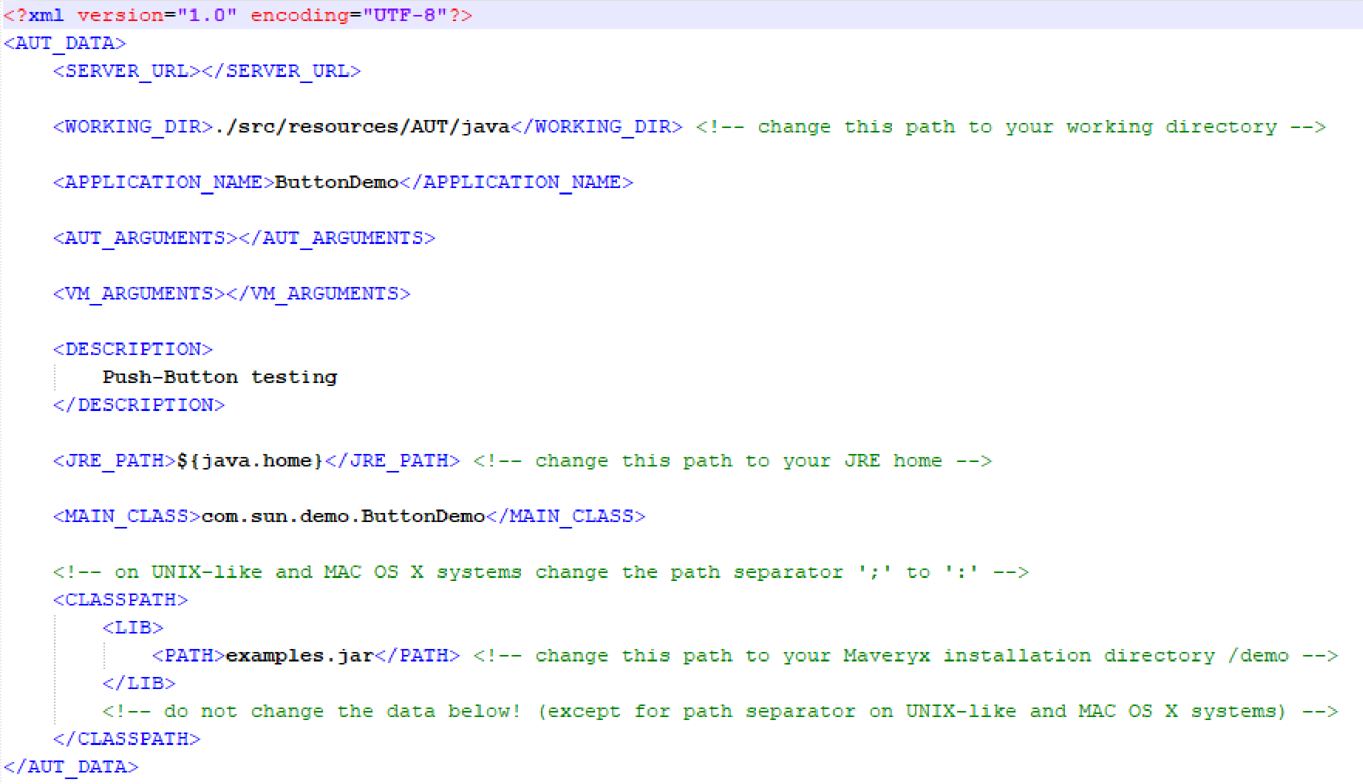
Cap III

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

Test it simple!

Java AUT Launch File

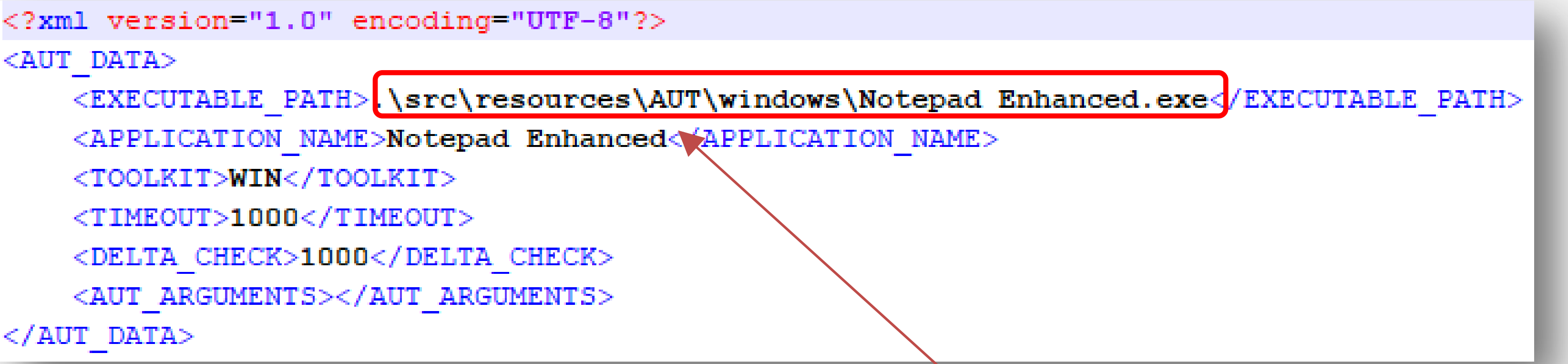
To execute a Java Application-Under-Test it is necessary to create the related **AUT launch file**.



```
<?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**.

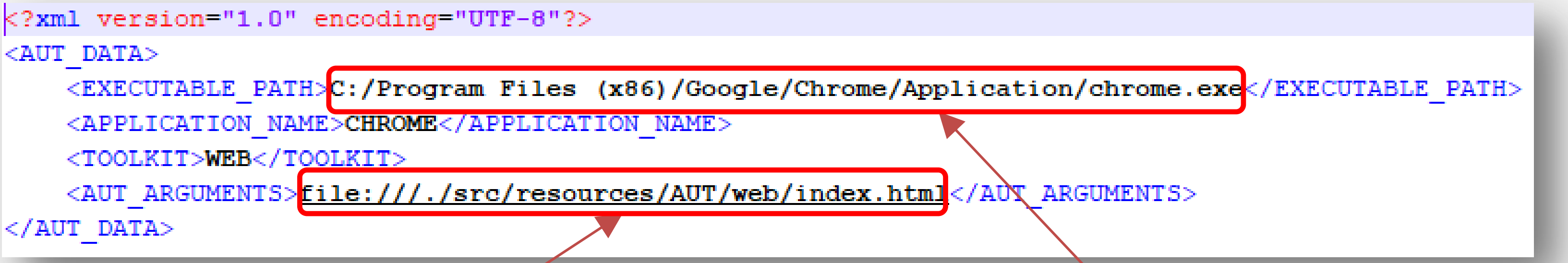


```xml
<?xml version="1.0" encoding="UTF-8"?>
<AUT_DATA>
    <EXECUTABLE_PATH>.\src\resources\AUT\windows\Notepad Enhanced.exe</EXECUTABLE_PATH>
    <APPLICATION_NAME>Notepad Enhanced</APPLICATION_NAME>
    <TOOLKIT>WIN</TOOLKIT>
    <TIMEOUT>1000</TIMEOUT>
    <DELTA_CHECK>1000</DELTA_CHECK>
    <AUT_ARGUMENTS>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

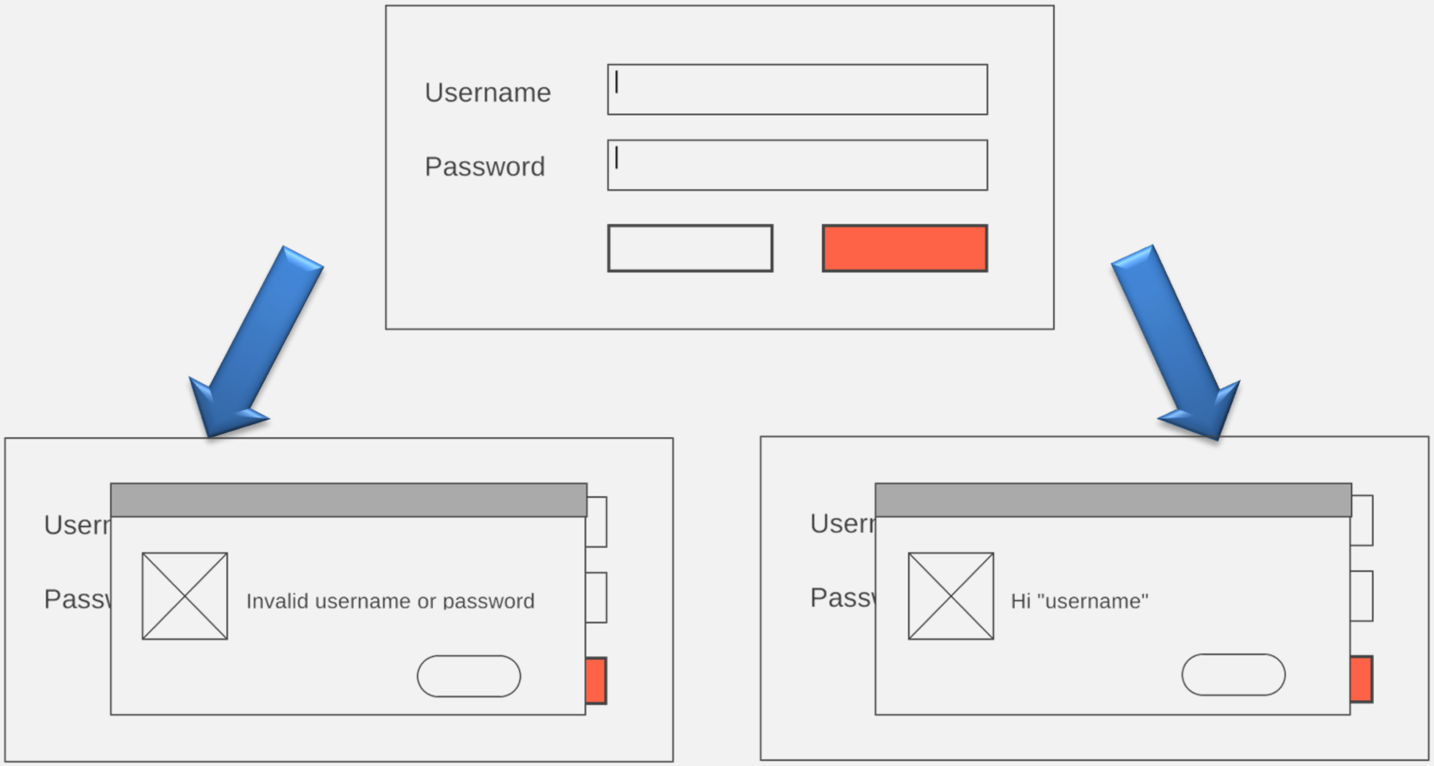


```
<?xml version="1.0" encoding="UTF-8"?>
<AUT_DATA>
    <EXECUTABLE_PATH>C:/Program Files (x86)/Google/Chrome/Application/chrome.exe</EXECUTABLE_PATH>
    <APPLICATION_NAME>CHROME</APPLICATION_NAME>
    <TOOLKIT>WEB</TOOLKIT>
    <AUT_ARGUMENTS>file:///./src/resources/AUT/web/index.html</AUT_ARGUMENTS>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

Set the URL of the AUT

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

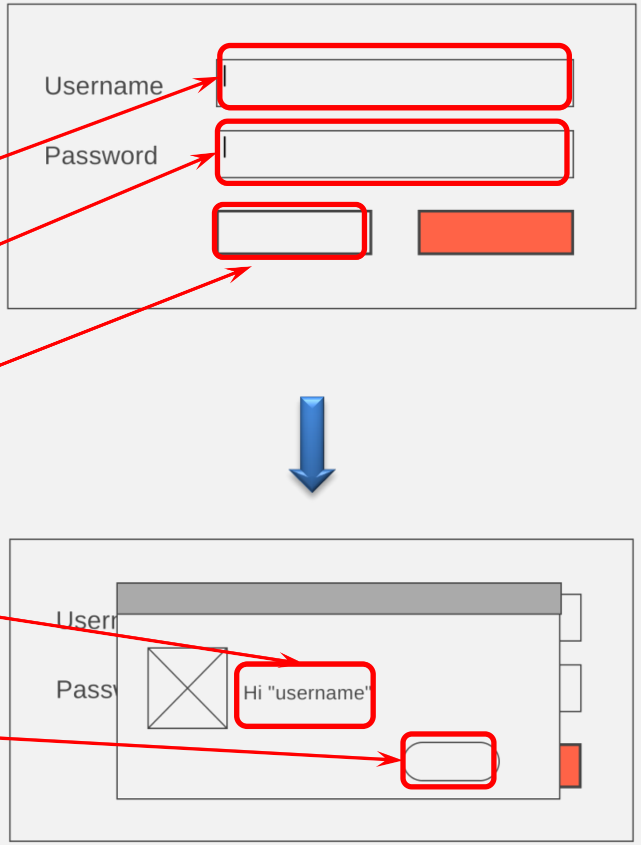
The Sample AUT



Test Case #001

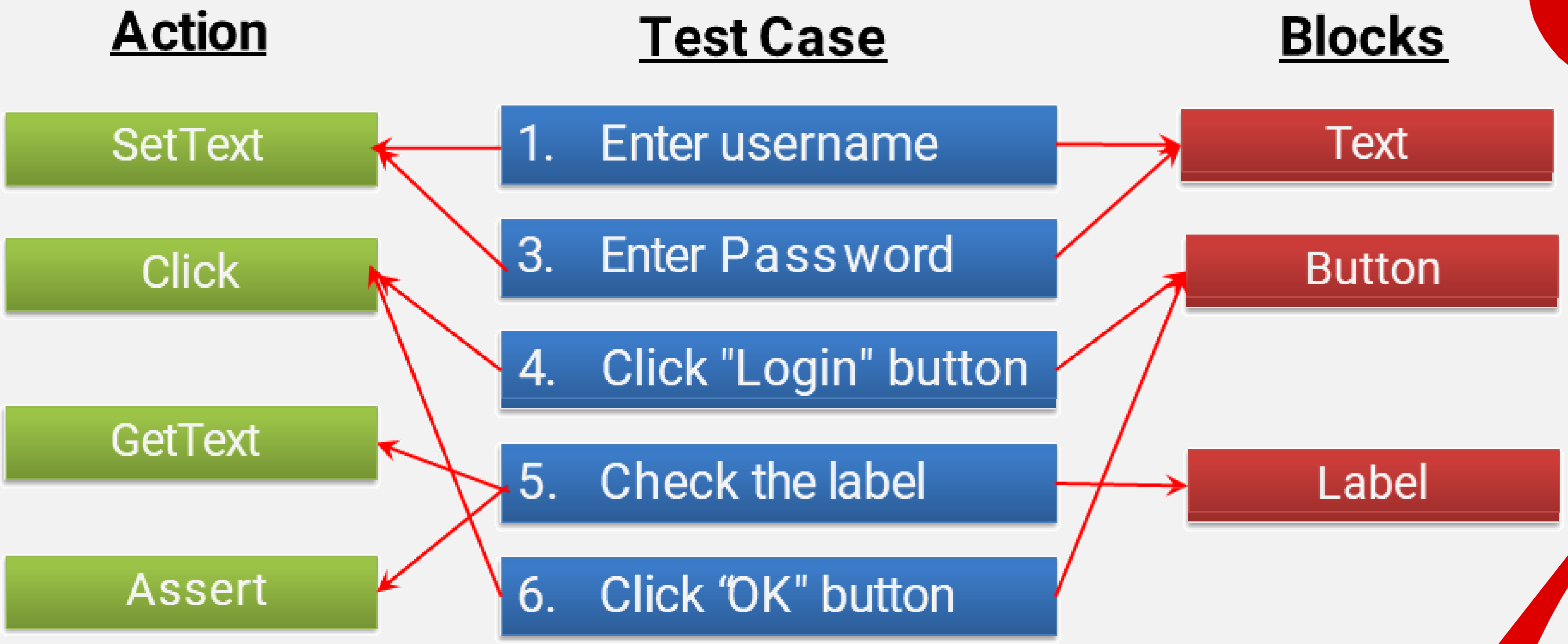
Test Case : TC_01

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



Test it simple!

Identify Blocks

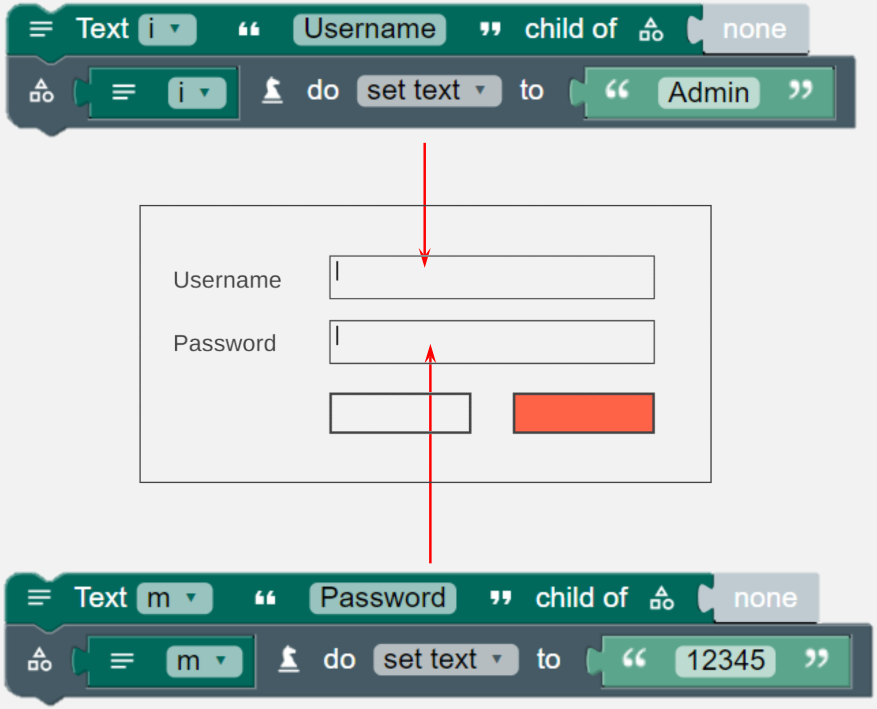


Test it simple!

Design Test Step 1 & 2

Test Case : TC_01

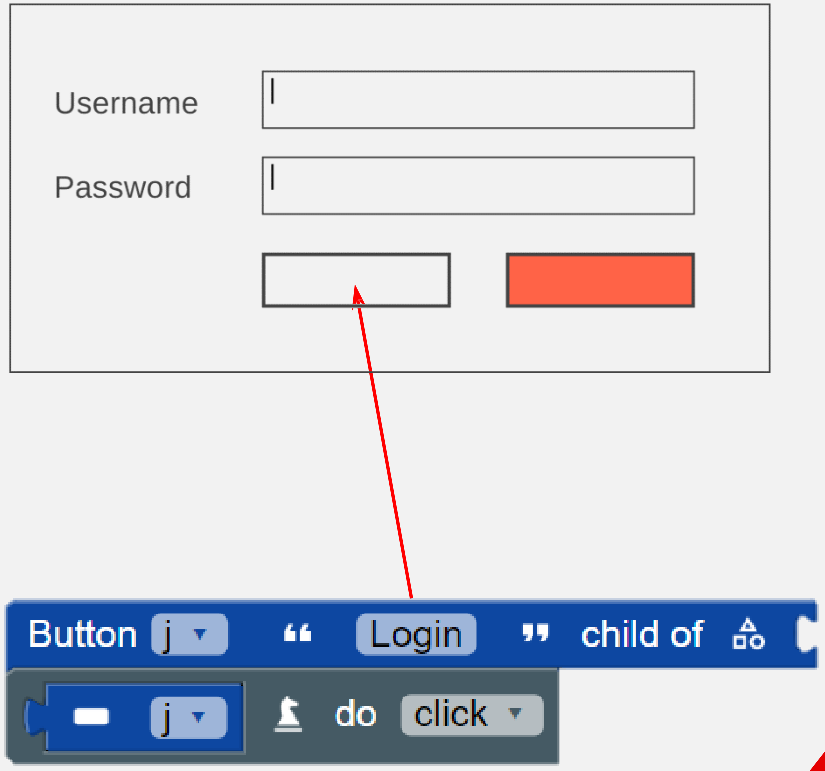
1. **Enter valid username**
2. **Enter valid Password**
3. Click "Login" button
4. Check label: “Hi Admin”
5. Click “OK" button


Text i “ Username ” child of none
i do set text to “ Admin ”
Username
Password
Text m “ Password ” child of none
m do set text to “ 12345 ”

Design Test Step 3

Test Case : TC_01

1. Enter valid username
2. Enter valid Password
3. **Click "Login" button**
4. Check label: “Hi Admin”
5. Click “OK" button

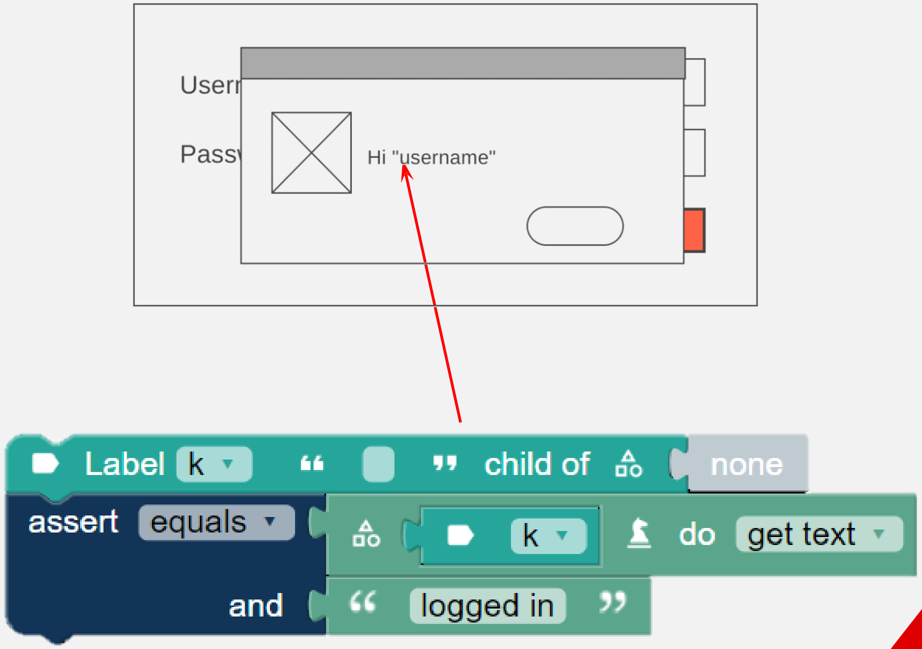


Test it simple!

Design Test Step 4

Test Case : TC_01

1. Enter valid username
2. Enter valid Password
3. Click "Login" button
4. **Check label: “Hi Admin”**
5. Click “OK" button



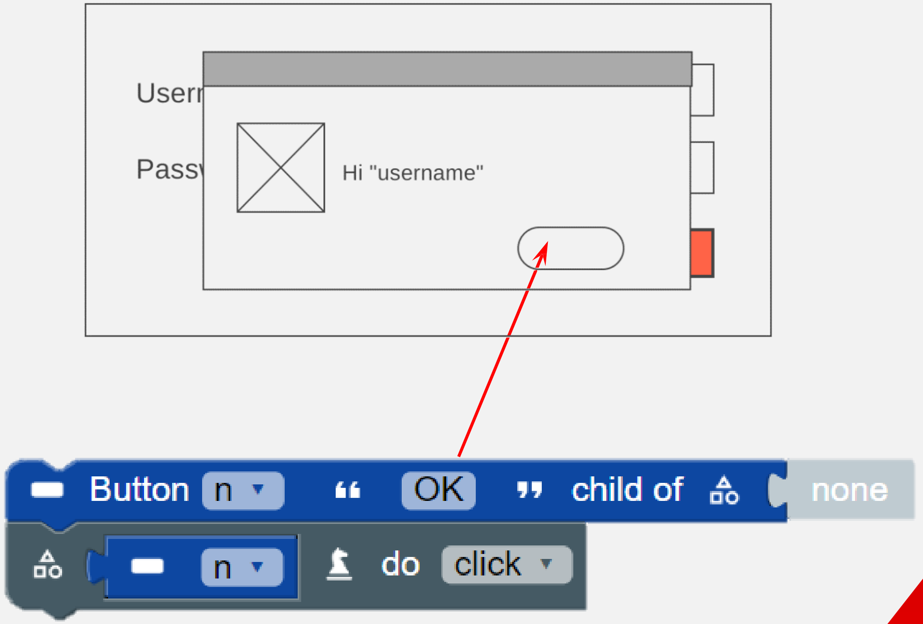
```
Label k " " child of none
assert equals  k do get text
          and  " logged in "
```

Test it simple!

Design Test Step 6

Test Case : TC_01

1. Enter valid username
2. Enter valid Password
3. Click "Login" button
4. Check label: “Hi Admin”
5. **Click “OK" button**



Test it simple!

THANK YOU

sales@maveryx.com

info@maveryx.com

+39 333 30 72 597
+39 351 87 85 706

MAVERYX®
Test it simple!