Calculator and Script Editing

Contents

Requirements

To follow along, and to understand calculators you will need:

- ✓ SYSTEM8 Ultimate
- ✓ Basic knowledge of TestFlow & Scripting

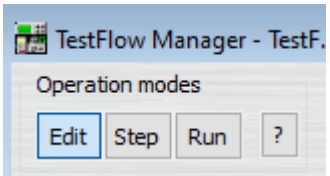
What you should know about Calculator

Calculator takes SYSTEM 8 Ultimate users operating TestFlows to the next level, by allowing the users to automatise the TestFlows, create multiple conditions, make calculations with the readings taken by the hardware, advance instruments setup from the calculator or create advance Reports.

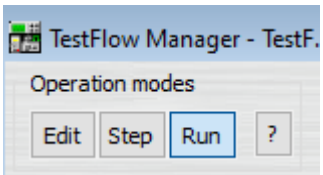
The calculator uses the simple, but flexible, programming language FormulaPlus.

Before creating the first script users should know:

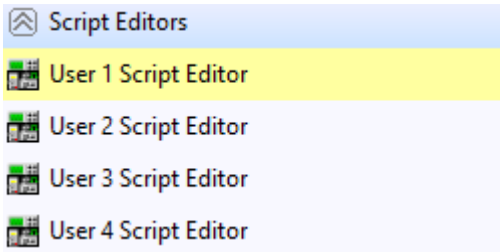
- When developing a script, within the *Operation modes* in the TestFlow, *Edit* mode must be selected.



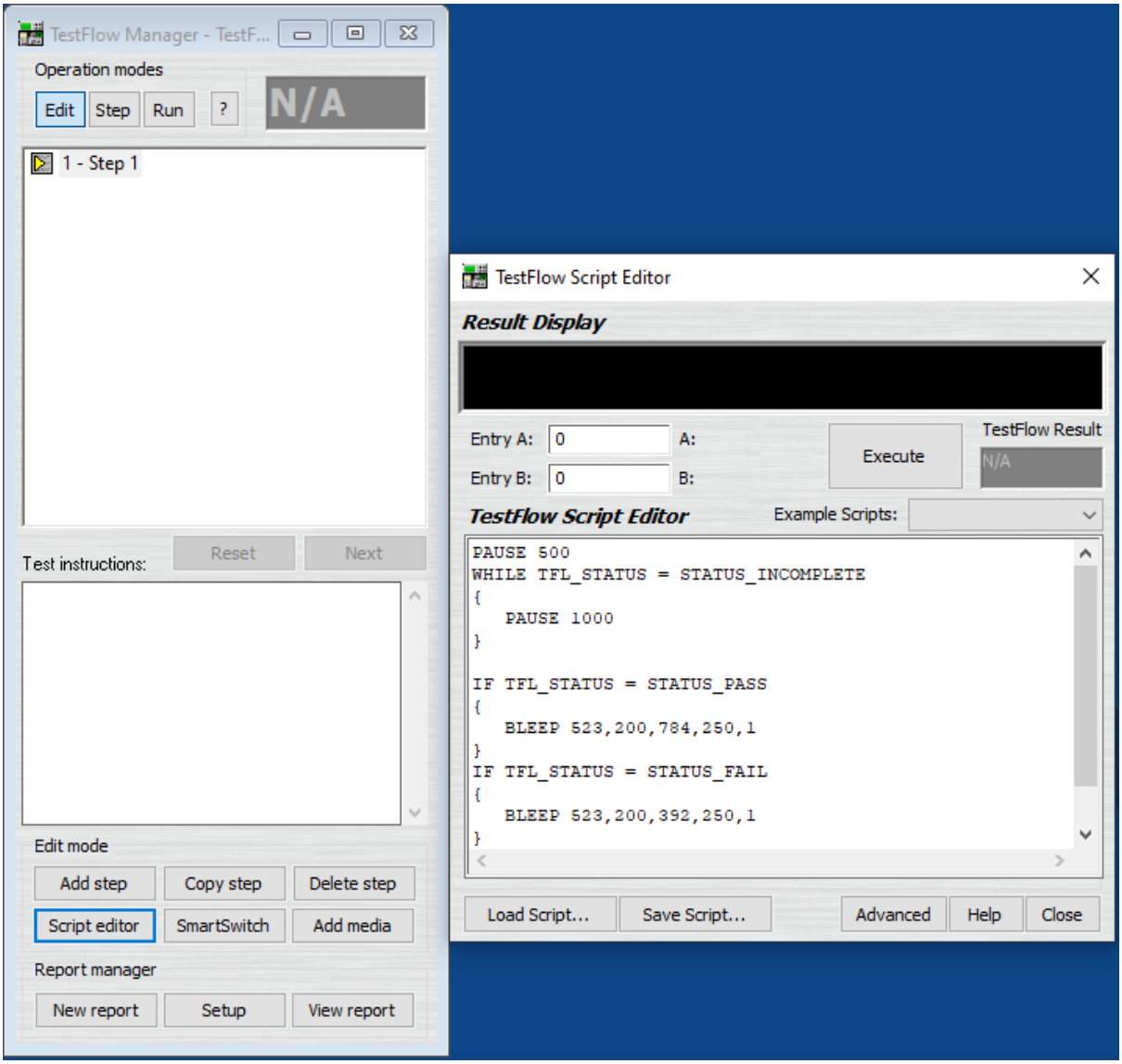
- Within the *Operation modes* in the TestFlow, when the user wants to run the script, *Run* mode must be selected.



- The SYSTEM 8 Ultimate software contains 9 user programmable calculators, four that are general purpose, and five that are related to individual instruments. Each calculator has its own readout display, comparison, and statistics functions.



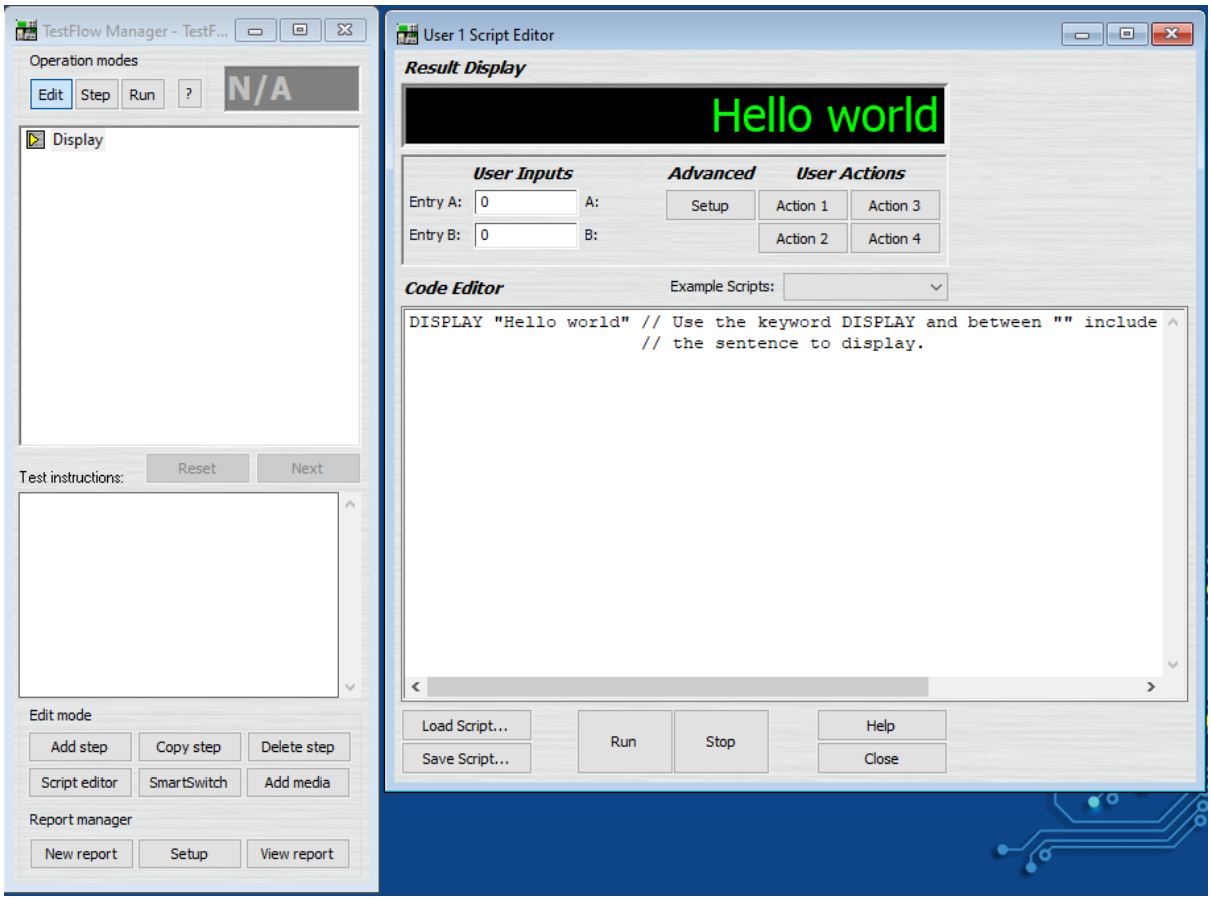
- Multiple calculators can be used in the same step allowing the user to have different scripts working at the same time.
- Calculator features are design to be used within the TestFlow. However, they also work outside of the TestFlow as well.
- User should be aware that each step of the TestFlow comes with a standard script already included in the TestFlow Script editor. In order to avoid any conflict between the existing Script and the new script check the existing Script and delete it if necessary.


TestFlow Manager - TestF...
Operation modes
Edit
Step
Run
?
N/A
1 - Step 1
Test instructions:
Reset
Next
Edit mode
Add step
Copy step
Delete step
Script editor
SmartSwitch
Add media
Report manager
New report
Setup
View report
TestFlow Script Editor
Result Display
Entry A:
0
A:
Entry B:
0
B:
Execute
TestFlow Result
N/A
TestFlow Script Editor
Example Scripts:
PAUSE 500
WHILE TFL_STATUS = STATUS_INCOMPLETE
{
PAUSE 1000
}
IF TFL_STATUS = STATUS_PASS
{
BLEEP 523,200,784,250,1
}
IF TFL_STATUS = STATUS_FAIL
{
BLEEP 523,200,392,250,1
}
Load Script...
Save Script...
Advanced
Help
Close

Script examples

Operating the display

Description: In this example we display a sentence, in this case “Hello world” using the Display.

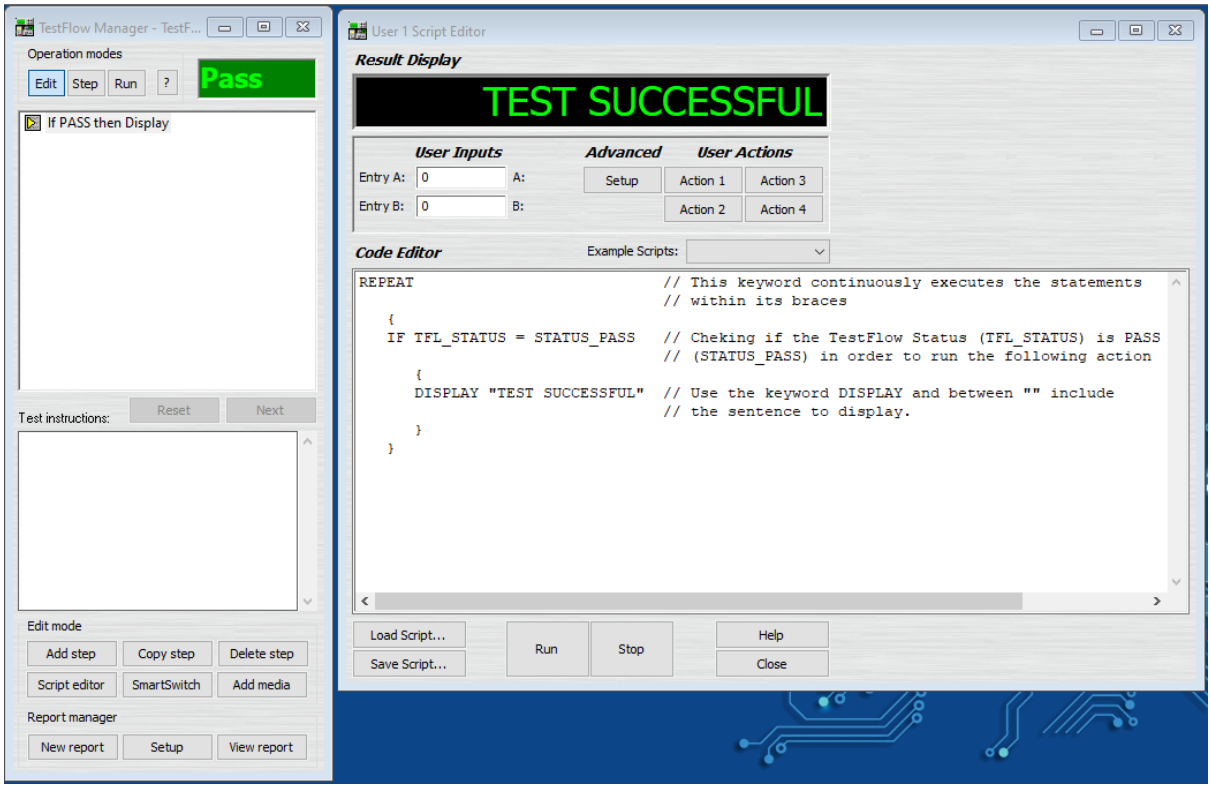


Script:

DISPLAY "Hello world" // Use the keyword DISPLAY and between " " include the sentence to display.

Checking if the TestFLow is PASS and using the display

Description: This example will work when the TestFlow gets a PASS when running a comparison test with one of the instruments. Then with the Script, we check if there is a PASS and if there is, the display will show “Test successful”

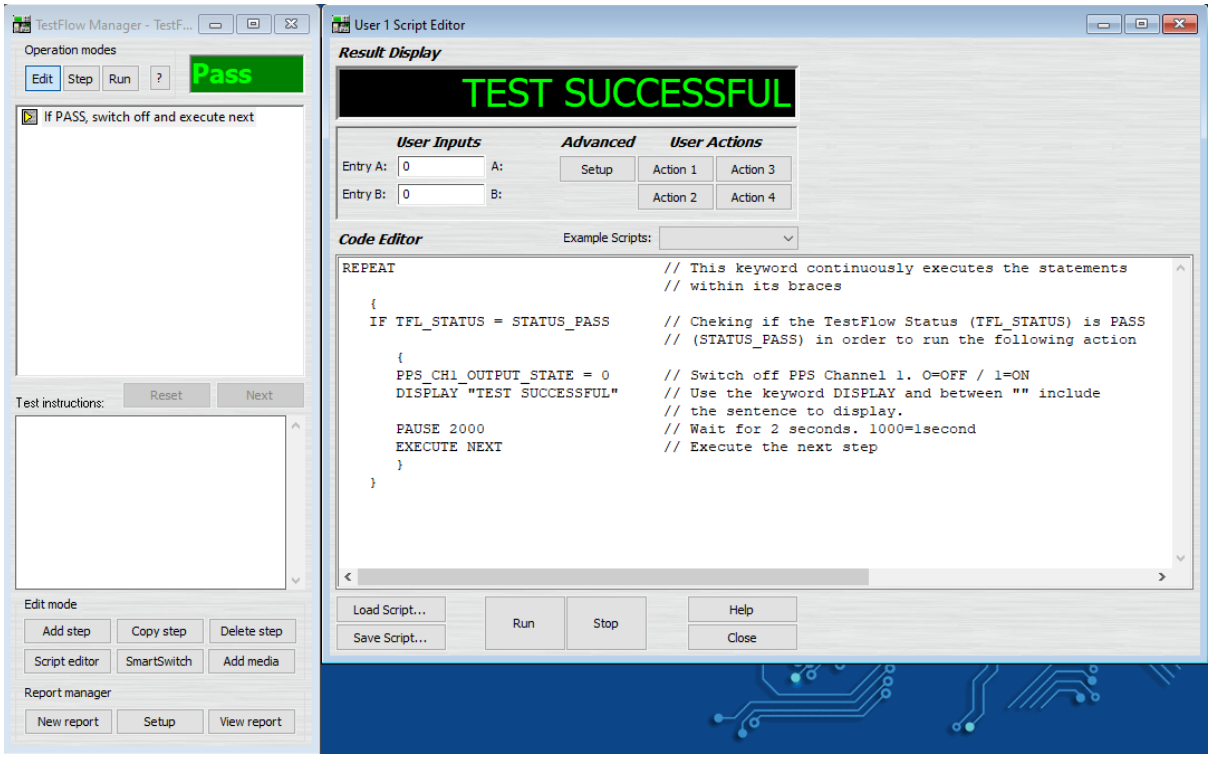


Script:

```
REPEAT                                  // This keyword continuously executes the statements
                                        // within its braces
        {
        IF TFL_STATUS = STATUS_PASS     // Checking if the TestFlow Status(TFL_STATUS) is PASS
                                        // (STATUS_PASS) in order to run the following action
                {
                DISPLAY "TEST SUCCESSFUL"   // Use the keyword DISPLAY and between "" include
                                        // the sentence to display.
                }
        }
```

Switch off the PPS and moving to the next step

Description: This example will work when the TestFlow gets a PASS when running a comparison test with one of the instruments. Then with the Script, we check if there is a PASS and if there is, channel 1 of the PPS will switch off automatically,

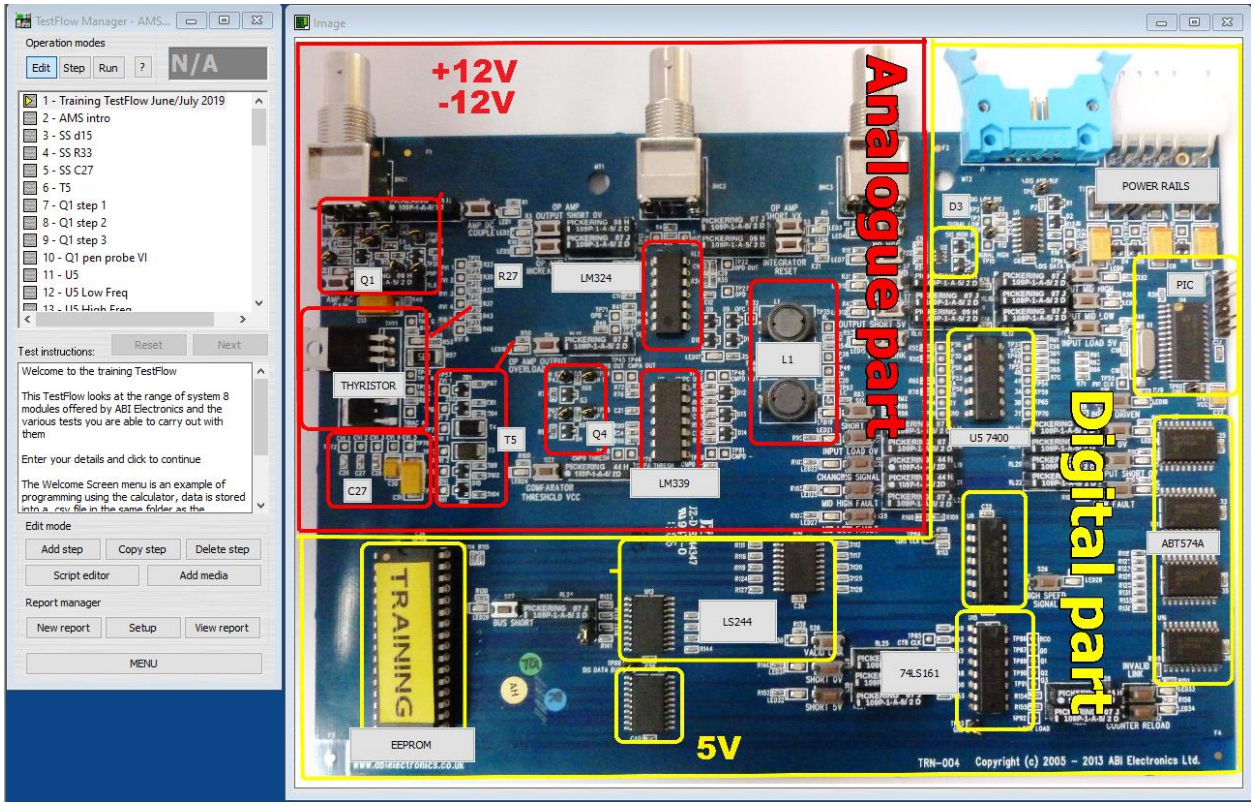


Script:

```
REPEAT                                  // This keyword continuously executes the statements
                                        // within its braces
        {
        IF TFL_STATUS = STATUS_PASS     // Checking if the TestFlow Status(TFL_STATUS) is PASS
                                        // (STATUS_PASS) in order to run the following action
                {
                PPS_CH1_OUTPUT_STATE = 0 // Switch off PPS Channel 1. O=OFF / 1=ON
                DISPLAY "TEST SUCCESSFUL" // Use the keyword DISPLAY and between "" include
                                        // the sentence to display.
                PAUSE 2000              // Wait for 2 seconds. 1000=1second
                EXECUTE NEXT            // Execute the next step
                }
        }
```

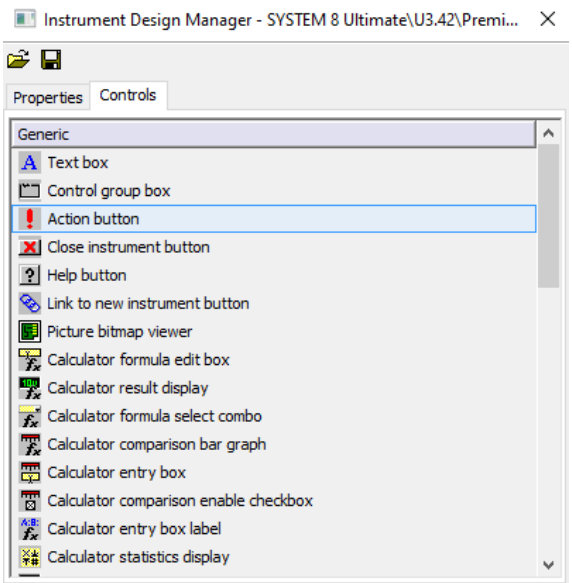
Main menu buttons

Description: In this example, first we need to create our Menu consisting of a picture and some *Actions buttons* which will take us to the step related to that button.
With the script, we make the action of jumping to a particular step possible.



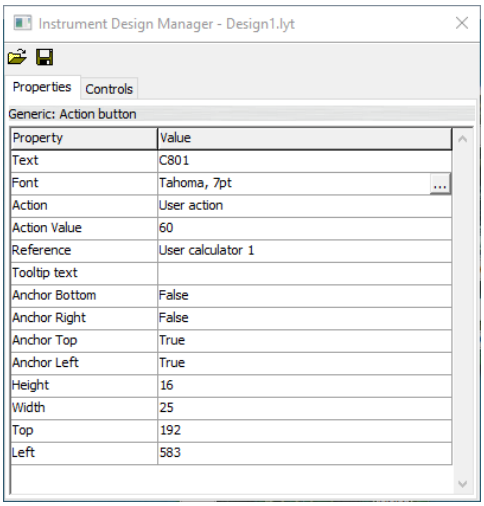
To create a menu such as the one picture above, firstly add an image of the PCB you wish to test. Then choose *Edit current instrument* in the Instruments drop-down menu. Add as many action buttons as you like.

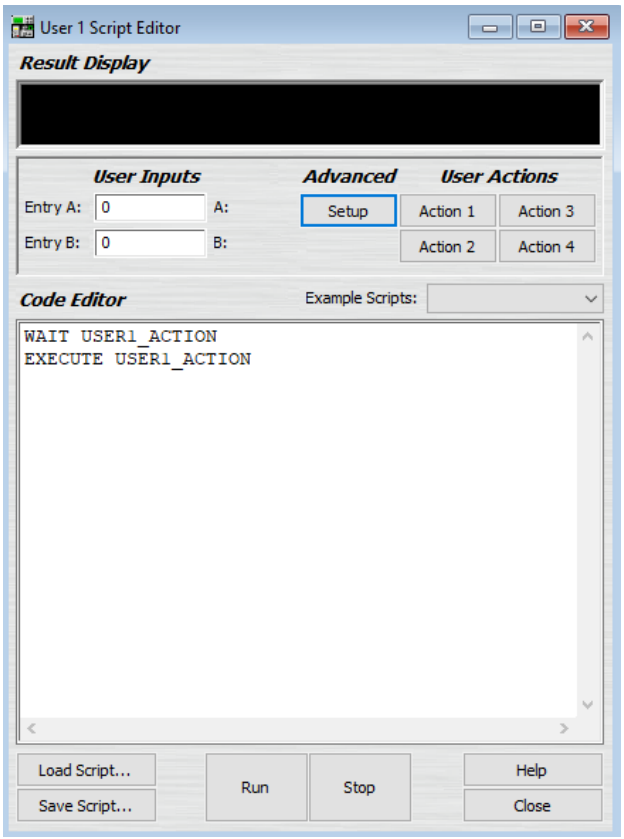
The action button can be found under the *Controls* tab, *Action button*



The action button needs to be setup as seen on the image below. In order to do so, click on top of your new action button and you will have access to the options can be found on the image below:

- Change the *Text* of the button which will be shown on the Menu
- The *Action Value* will be related to the step (step 60) which this button will be related to.
- Make sure the *Reference* is related to the Calculator that will be use for this menu. In this case *User calculator 1* for the *User 1 Script Editor* Instrument.





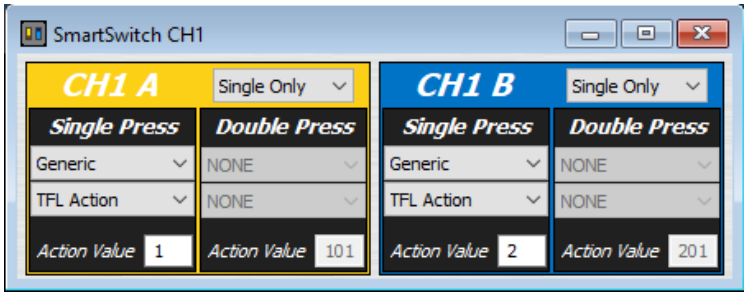
***Script*:**

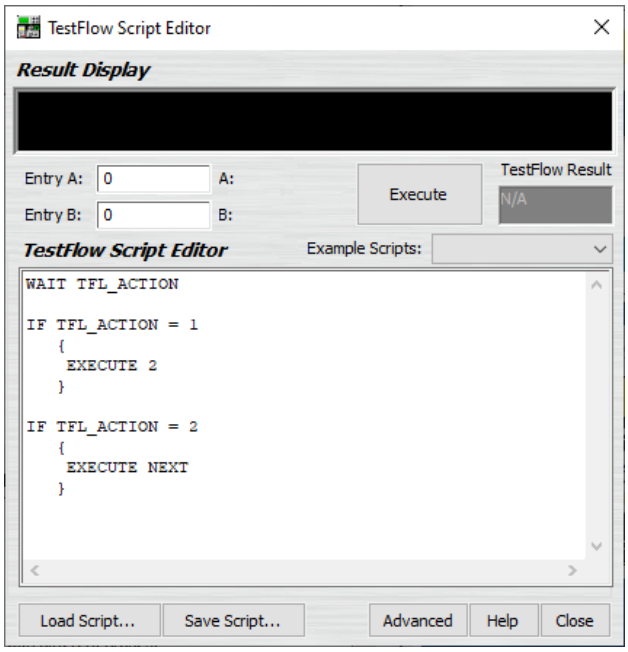
WAIT USER1_ACTION //The calculator is wait for the user to click on an action button

EXECUTE USER1_ACTION //The TestFlow will execute the step included in the Action Value

Integrating the SmartSwitch with scripts

Description: This script uses the SmartSwitch to activate custom functions. In this example, the script is used to make the SmartSwitch take the operator to particular steps in the TestFlow. The SmartSwitch has been set up with Action Values for Channel A (Action Value = 1) and B (Action Value = 2) and to be used with the TestFlow Script Editor (TFL).



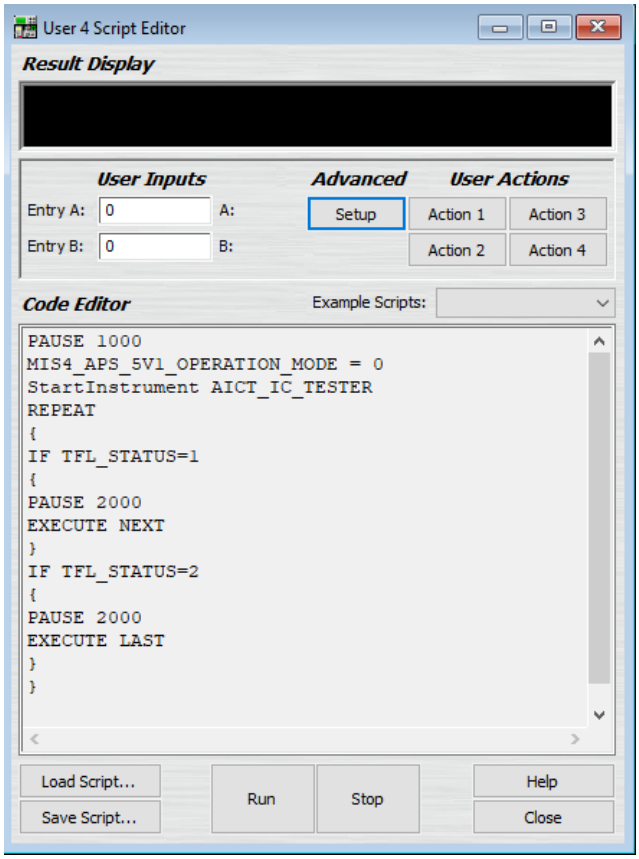


Script:

```
WAIT TFL_ACTION                          // Wait until the operator
                                         // performs an action
IF TFL_ACTION = 1                        // When the action '1' is
                                         // performed move to step two
        {
         EXECUTE 2                       // Move to step two
        }

IF TFL_ACTION = 2                        // When the action '2' is
                                         // performed move to next step
        {
         EXECUTE NEXT                    // Move to the next step
        }
```

Automatically start an instrument

Description: This script automatically starts the AICT Analogue IC Tester instrument automatically, before that it turns off the APS and after the instrument starts automatically, it checks if the TestFlow gives PASS or FAIL and it jump into the next step or the last step.

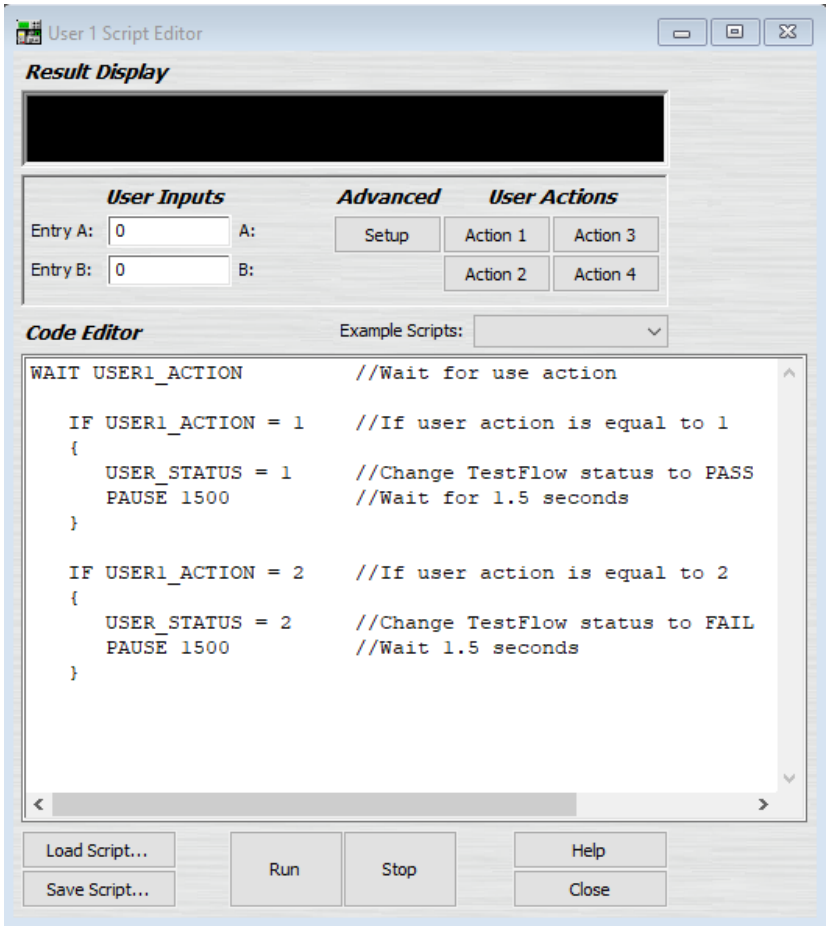


Script:

```
PAUSE 1000                                  //Wait for 1 second
MIS4_APS_5V1_OPERATION_MODE = 0             //Turn the 5V APS from the MIS4 off
StartInstrument AICT_IC_TESTER              //Start the test
REPEAT                                      //Repeat the script
{
IF TFL_STATUS=1                             //If TestFlow passes
{
        PAUSE 2000                          //Wait 2 seconds
        EXECUTE NEXT                        //Move onto next step
}
IF TFL_STATUS=2                             //If TestFlow fails
{
        PAUSE 2000                          //Wait 2 seconds
        EXECUTE LAST                        //Jump to last step
}
}
```

Changing the TestFlow status to pass/fail

Description: This script looks at how you can force the TestFlow to display a pass/fail using an action button. In this case we use the actions buttons that comes on the Script Editor, button “Action 1” = 1 and button “Action 2” = 2.

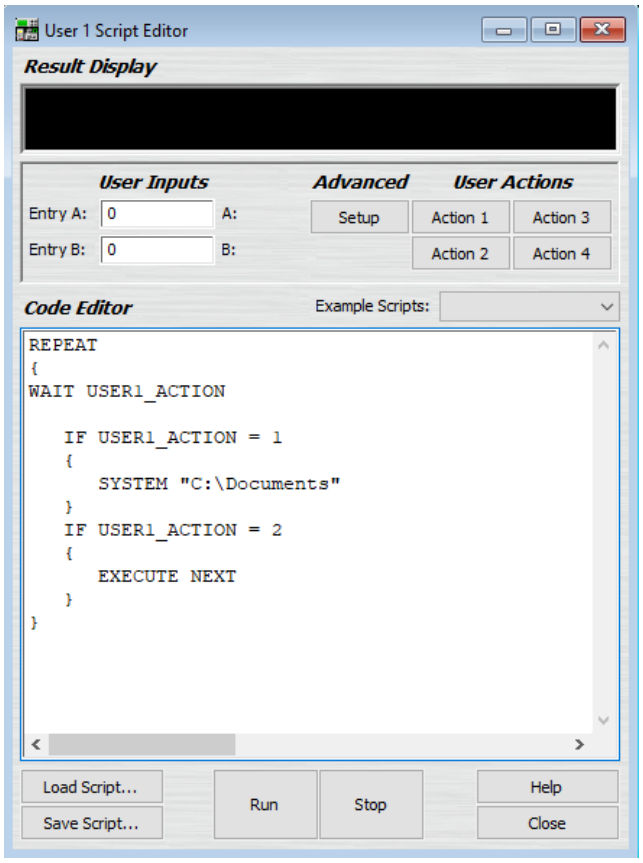


Script:

```
WAIT USER1_ACTION                    //Wait for use action

        IF USER1_ACTION = 1          //If user action is equal to 1
        {
                USER_STATUS = 1      //Change TestFlow status to PASS
                PAUSE 1500           //Wait for 1.5 seconds
        }

        IF USER1_ACTION = 2          //If user action is equal to 2
        {
                USER_STATUS = 2      //Change TestFlow status to FAIL
                PAUSE 1500           //Wait 1.5 seconds
        }
```

How to open an external document

Description: This script opens an external document from within TestFlow. In this case we use the actions buttons that comes on the Script Editor, button “Action 1” = 1 and button “Action 2” = 2.

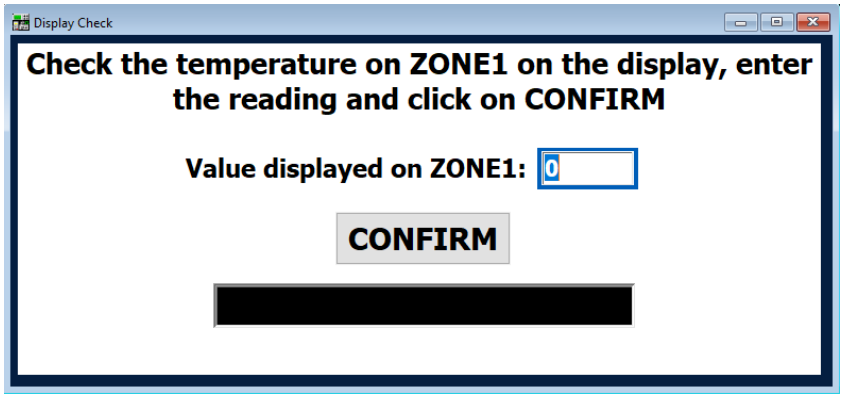


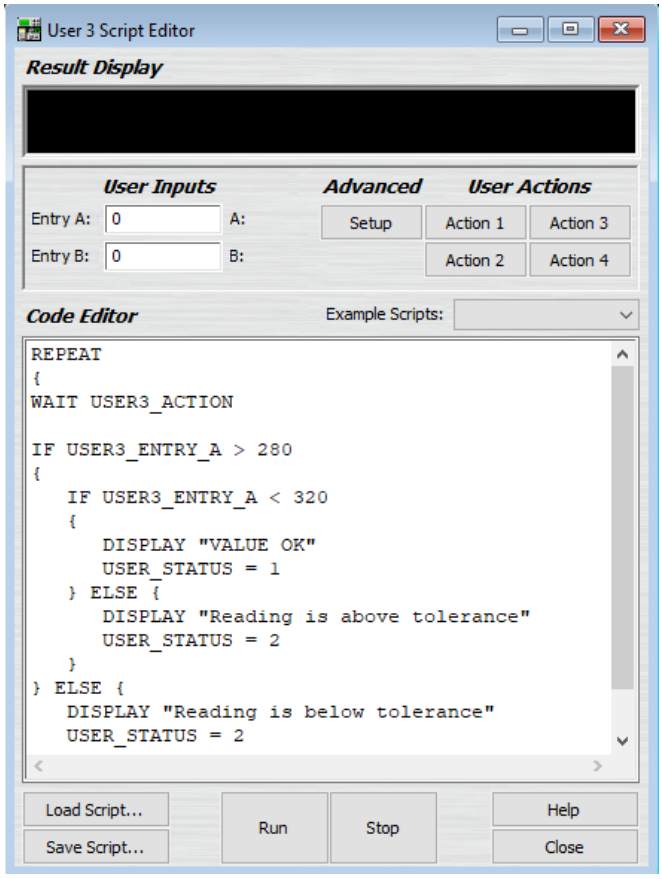
Script:

```
REPEAT                                  //Repeat the below script over and over
{
WAIT USER1_ACTION                       //Wait for a user action

        IF USER1_ACTION = 1             //If user action is equal to 1
        {
                SYSTEM "C:\Documents"   //Open up this file located in C drive
        }
        IF USER1_ACTION = 2             //If user action is equal to 2
        {
                EXECUTE NEXT            //Move onto the next step
        }
}
```

Operator inputting data for comparison

Description: This script makes use of a custom instrument as well as the script editor. The operator is asked to carry out a visual check and enter the value in the text box. Once the 'CONFIRM' button is pushed a comparison is done in the script editor, TestFlow is set to PASS or FAIL and some text is displayed. The text box is a "Calculator entry box" and the 'CONFIRM' an Action button.





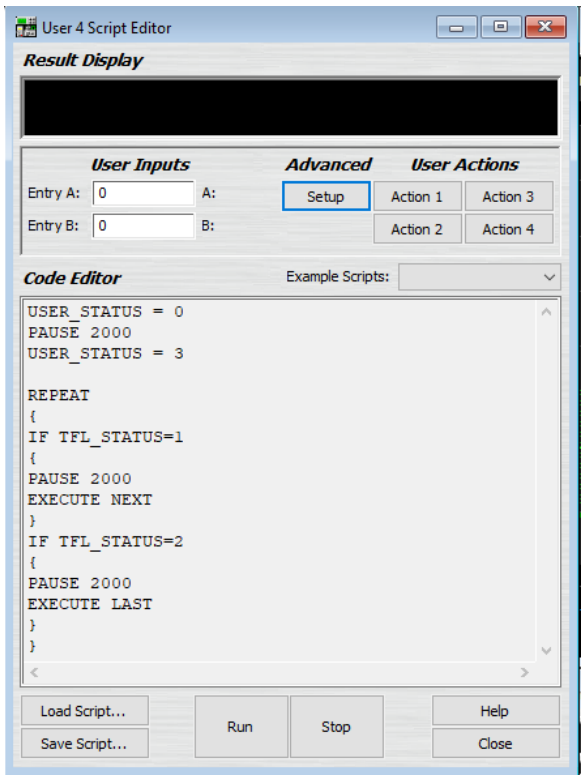
Script:

```
REPEAT                                  //Run this script over and over again
{
WAIT USER3_ACTION                       //Wait for button to be pressed

IF USER3_ENTRY_A > 280                  //If the operator enters a number greater than 280
```

```
{
                    IF USER3_ENTRY_A < 320          //If the entered value is also less than 320
                    {
                       DISPLAY "VALUE OK"           //Display text
                       USER_STATUS = 1              //Change TestFlow status to PASS
                    } ELSE {                        //If the entered value is greater than 320
                       DISPLAY "Reading is above tolerance"   //Display text
                       USER_STATUS = 2              //Change TestFlow status to FAIL
                    }
} ELSE {                                             //otherwise carry out the following script below
                    DISPLAY "Reading is below tolerance" //Display text
                    USER_STATUS = 2                 //Change TestFlow status to FAIL
}
}
```

Delay before starting a test

Description: This script makes use of the PAUSE function which pauses execution, either displaying a prompt or waiting for a given number of milliseconds. This can be useful when you want to delay the TestFlow from doing a comparison so that the instruments and signals have all been loaded.

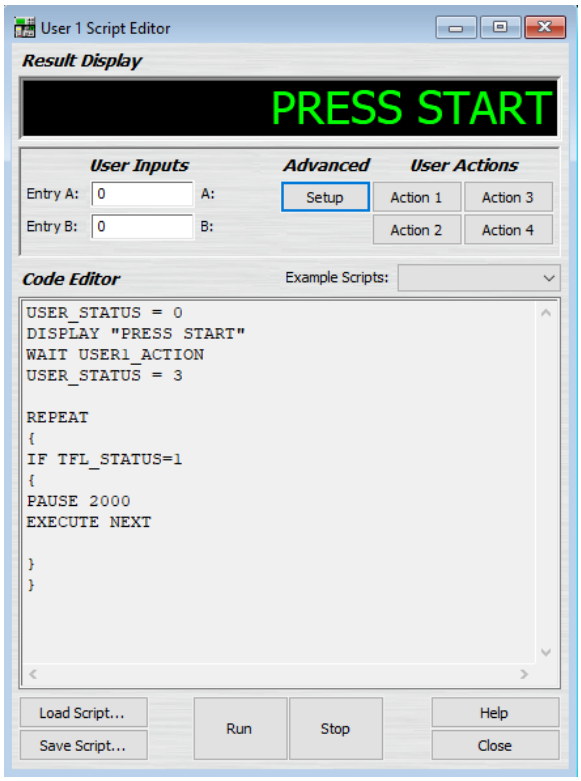


Script:

```
USER_STATUS = 0                    //Change TestFlow status to Incomplete
PAUSE 2000                         //Wait for 2 seconds
USER_STATUS = 3                    //Change TestFlow status to None

REPEAT                             //Run this script over and over again
{
IF TFL_STATUS=1                    //If the TestFlow is a pass
{
PAUSE 2000                         //Wait for 2 seconds
EXECUTE NEXT                       //Move to next step
}
IF TFL_STATUS=2                    //If TestFlow fails
{
PAUSE 2000                         //Wait for 2 seconds
EXECUTE LAST                       //Jump to last step
}
}
```

User action delay

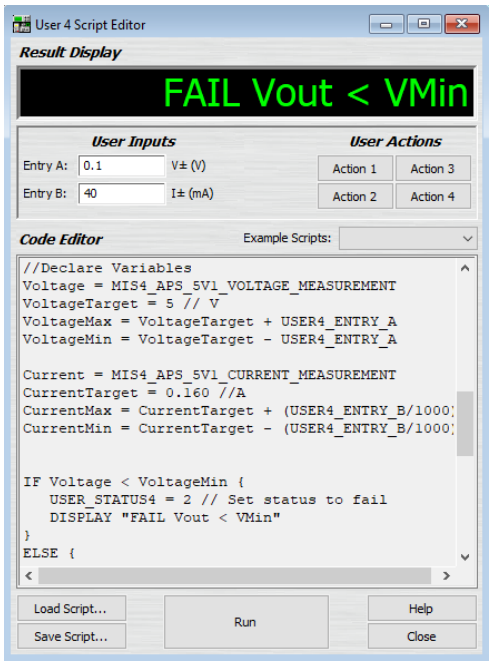
Description: In this script, the test remains dormant until the user selects the custom button. The button is assigned a user value and when the TestFlow receives the user value it will proceed with the test.




```
USER_STATUS = 0                    //Change TestFlow status to Incomplete
DISPLAY "PRESS START"
WAIT USER1_ACTION                  // Waits for the user to press the
                                   // button
USER_STATUS = 3                    //Change TestFlow status to None

REPEAT                             //Run this script over and over again
{
IF TFL_STATUS=1                    // Waits for a pass until it can move to
                                   // the next step
{
PAUSE 2000                         //Wait for 2 seconds
EXECUTE NEXT                       //Move to next step
}
}
```

Using a reading from an instrument for comparison

Description: This script takes a voltage, and current reading from the Auxillary Power Supply from the MIS4 and compares that against what it should be outputting. The script editor also uses the tolerances defined by the user in the Entry A/B text boxes. It then compares the measured result against the target and displays the correct response.



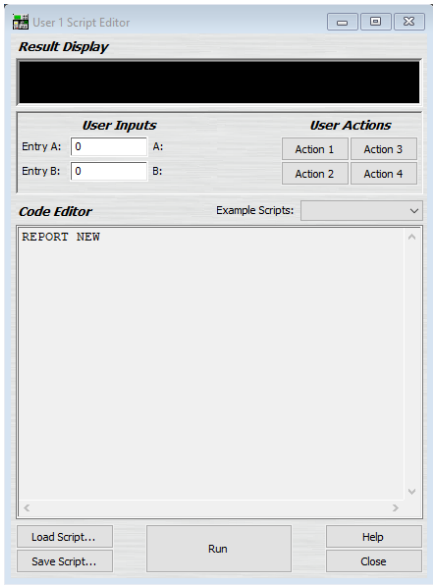
Script:

```
//Declare Variables
Voltage = MIS4_APS_5V1_VOLTAGE_MEASUREMENT        // Takes the voltage measurement and assigns
                                                  // it to the variable Voltage
VoltageTarget = 5                                 // The target voltage for the test
VoltageMax = VoltageTarget + USER4_ENTRY_A        // Plus and minus the tolerance decided
VoltageMin = VoltageTarget - USER4_ENTRY_A

Current = MIS4_APS_5V1_CURRENT_MEASUREMENT        // Measurement for the current
CurrentTarget = 0.160                             // Target current
CurrentMax = CurrentTarget + (USER4_ENTRY_B/1000) // Target current plus the tolerance
CurrentMin = CurrentTarget - (USER4_ENTRY_B/1000)


IF Voltage < VoltageMin {                         // Checks whether the measurement is
                                                  // less than than the target minimum
                                                  // tolerance
        USER_STATUS4 = 2                          // Set status to fail
        DISPLAY "FAIL Vout < VMin"                // Display for the calculator
}
ELSE {                                            // If the previous comparison fails then…
        IF Voltage > VoltageMax {                 // Check whether the voltage is greater than
```

```
                                                        // the maximum
                USER_STATUS4 = 2                        // Set status to fail
                DISPLAY "FAIL Vout > VMax"              // Display for calculator
        }
}

IF USER_STATUS4 = 0 {                                   // If user status is still Incomplete

        IF Current < CurrentMin {                       // Check whether the measurement lies within
                                                        // the tolerances
                USER_STATUS4 = 2                        // Set status to fail
                DISPLAY "FAIL Iout < IMin"
        }
        ELSE {
                IF Current > CurrentMax {               //Check whether the current is greater than
                        USER_STATUS4 = 2                // Set status to fail
                        DISPLAY "FAIL Iout > IMax"
                }
        }
}
IF USER_STATUS4 = 0 {                                   // If user status is still Incomplete

        USER_STATUS4 = 1                                //Set the TestFlow to PASS
        DISPLAY "V PASS I PASS"                         // Display for the calculator
}
```

Creating a report

Description: This script automatically opens the windows file explorer for the operator to find a destination to save a new report. It will happen as soon as the step starts.

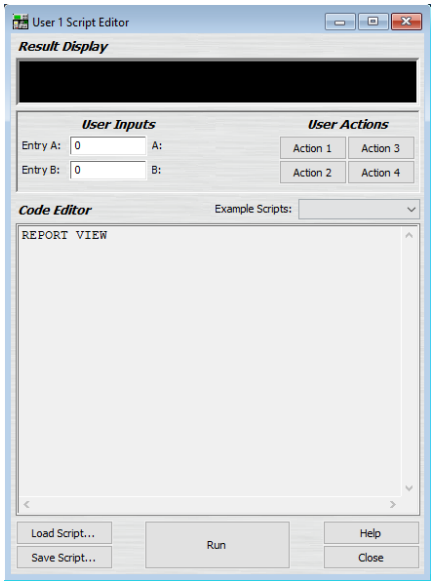


Script:

REPORT NEW //requests the user to create/save a new report

Opening a report

Description: This script directly opens the Report for the TestFlow. Our customers use this feature to display the Report at the end of their test sequences.

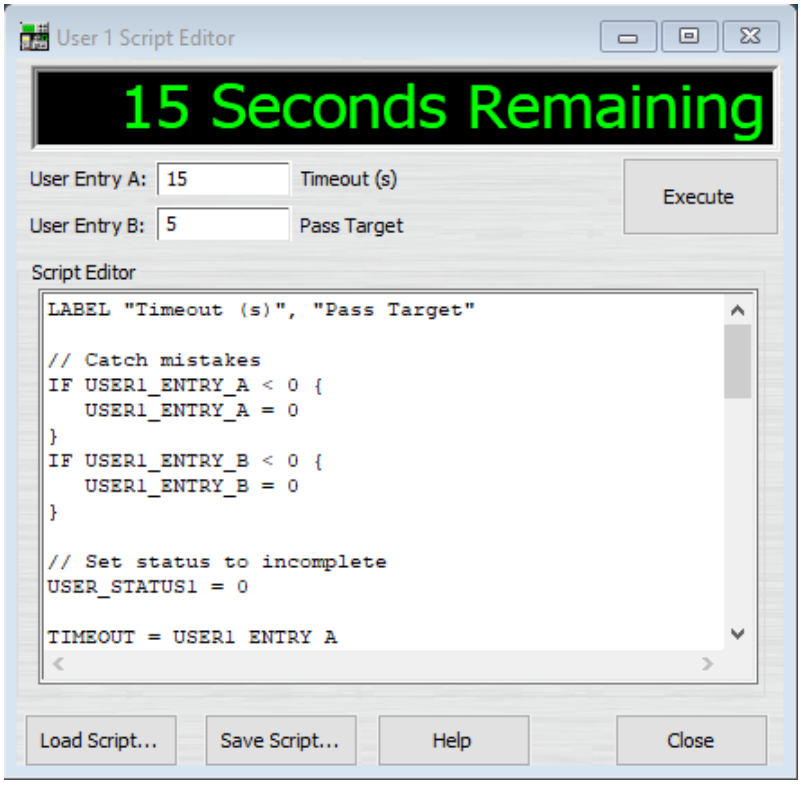


Script:

REPORT VIEW // Opens the report on this step

Countdown timer

Description: This script delays the overall TFL result by setting user status 1 to waiting for a given time out. The time out can be skipped if the instrument passes a given number of times in a row. This is useful for instruments that require manual setup or time to settle e.g. allowing DSO, DMM instruments to be reliably used in TFL run mode without extra user interaction such as clicking an action button. e.g. User Entry A = 10, User Entry B = 5 The code will wait for a maximum of 10 seconds or 5 passes in a row which ever happens first.



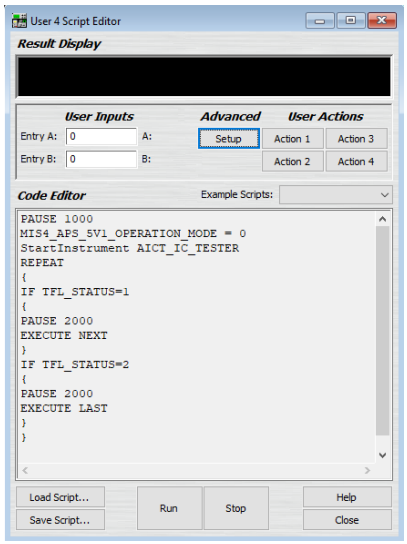
Script:

```
LABEL "Timeout (s)", "Pass Target"          //Set the text displayed in the in the calculator's
                                            // associated calculator entry box label control
IF USER1_ENTRY_A < 0 {                                  // This is a comparison to make sure
                                                        // the user hasn't incorrectly entered
                                                        // wrong values
        USER1_ENTRY_A = 0
}
IF USER1_ENTRY_B < 0 {                      //Check that the value entered is not negative
        USER1_ENTRY_B = 0                   //Sets the value to 0 if the value entered was
                                            //negative
}

USER_STATUS1 = 0                                        // Set status to incomplete

TIMEOUT = USER1_ENTRY_A                                 // Adds the user defined Entrys to
```

```
                                                        // variables
PASS_COUNT_TARGET = USER1_ENTRY_B
PASS_COUNT = 0
FORMAT 0, " Seconds Remaining", N                       // This changes the format of the
                                                        // readout. In this case 0 desimal
                                                        //places, with Seconds Remaining
                                                        //displayed as the unit and displayed
                                                        //as a regular number
WHILE TIMEOUT > 0 {                                     // It will countdown as long as the
                                                        // countdown is greater than zero
        DISPLAY TIMEOUT

        PAUSE 1000                                      // Wait 1 second

        IF USER1_ENTRY_B > 0 {                          // Check if we have a pass count of
                                                        // more than 0
                IF MIS4_DOM1_COMPARISON_RESULT = 1 {    // Check if the result is passing
                        PASS_COUNT = PASS_COUNT + 1     // Decrement the pass count
                        IF PASS_COUNT > PASS_COUNT_TARGET {// Check if we have enough pass'
                                TIMEOUT = 0             // Skip the rest of the timeout
                                DISPLAY "Pass Count Reached"
                                PAUSE 2000              // Wait 1 second
                        }
                }
                ELSE {
                        PASS_COUNT = USER1_ENTRY_B      // Reset the pass count
                }
        }
        TIMEOUT = TIMEOUT – 1                           // Decrement the timeout
}
DISPLAY " "                                             // Clear the display
USER_STATUS1 = 3                                        // Set status back to NA
```

Start Instrument

Description: In this example, the script editor has started the test automatically. Our customers use this example for their bed of nails solutions because it automatically starts the instruments when required.



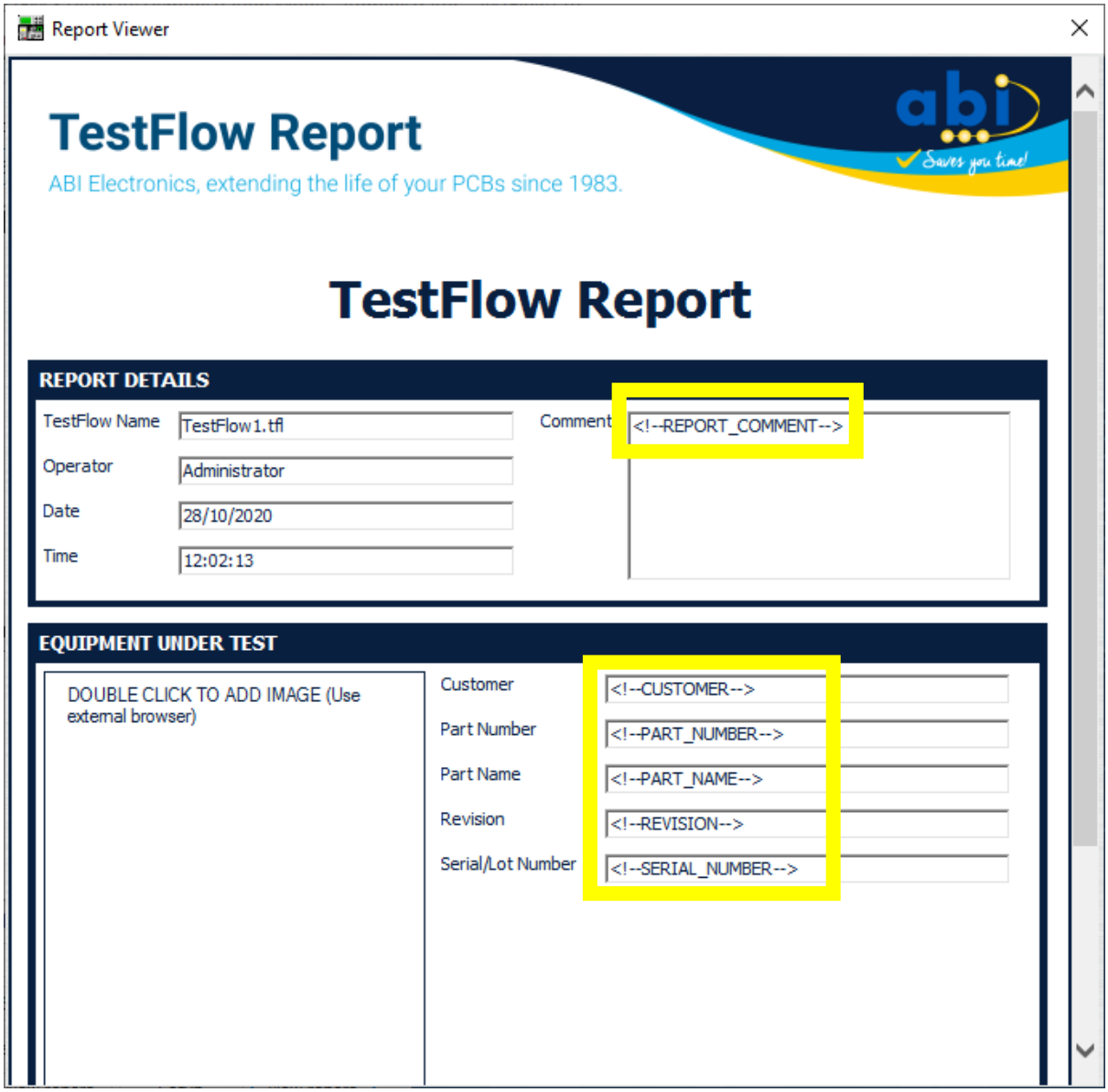
Script:

```
PAUSE 1000                                  // Wait one second
MIS4_APS_5V1_OPERATION_MODE = 0             // This turns off the power supply for
                                            // the MIS4 before starting the test
StartInstrument AICT_IC_TESTER              // Starts the IC Tester instrument
REPEAT                                      // Continuously checks whether the
                                            // instrument passes or fails
{
IF TFL_STATUS=1                             // If pass move on to the next step
{
PAUSE 2000                                  // Wait 2 second
EXECUTE NEXT                                //Move to next step
}
IF TFL_STATUS=2                             // If fail move on to the last step
{
PAUSE 2000                                  // Wait 2 second
EXECUTE LAST                                //Jump to last step
}
}
```

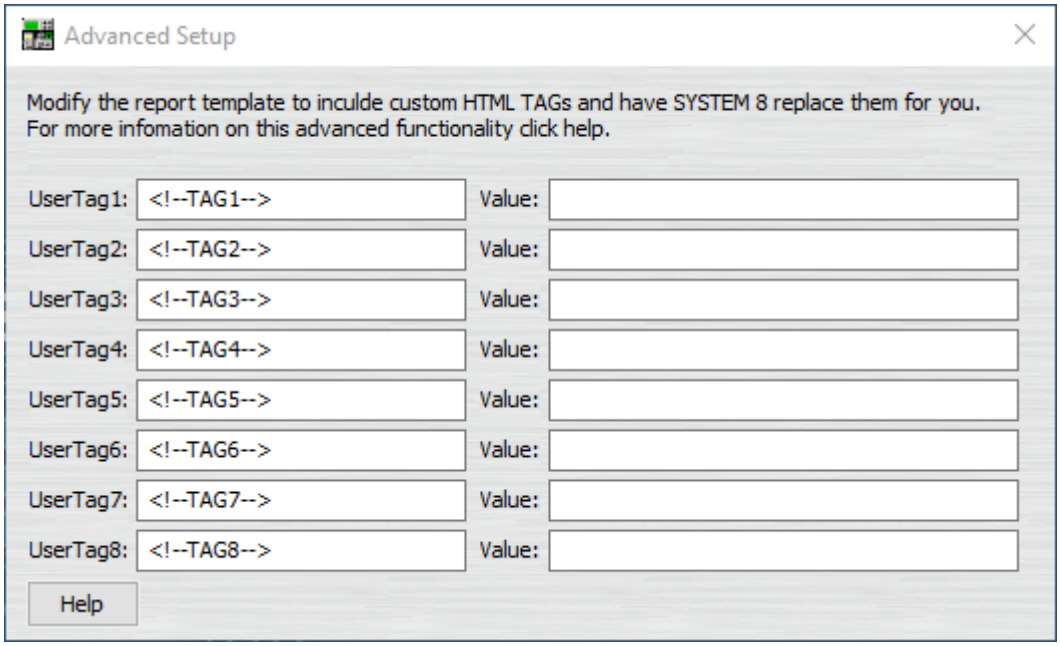
Advanced Features

Description: In the most recent version of SYSTEM8 Ultimate, we have added the ability to add TAGS to your report. To access this you need to go to Report: Setup then to Advanced and you can match the TAGS to the ones on the report.

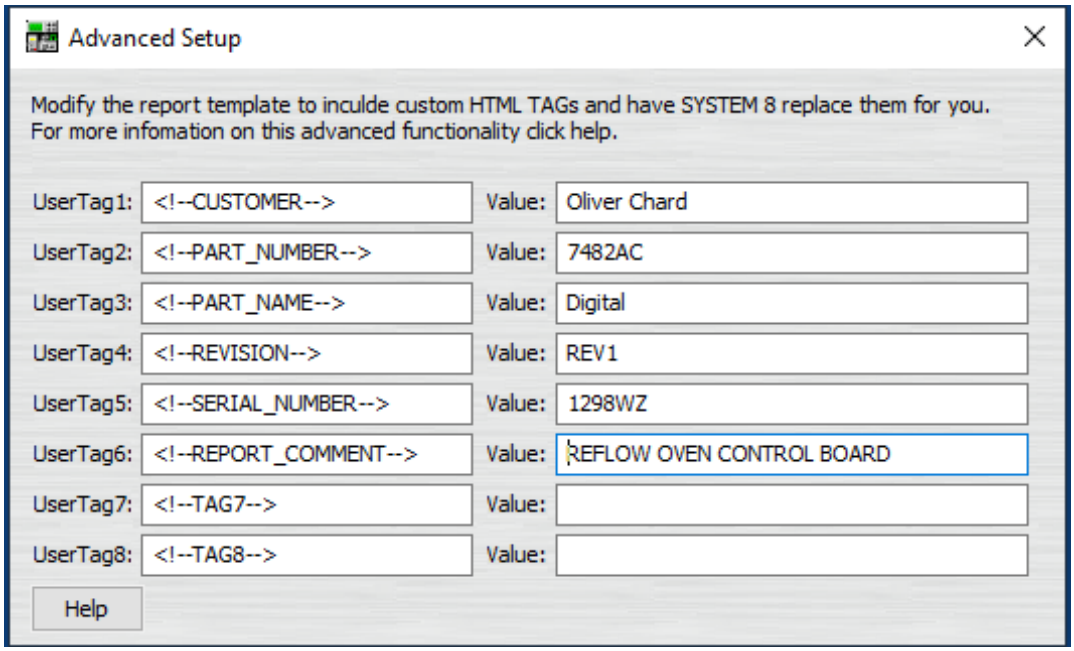
No tags used in the report. This is the report that comes by default:



And this is the Advanced Setup that comes by default:



Once tags have been introduced:



Once the tags have been replaced within the report 'Advanced Setup' window, you will see the desired information within the report itself as shown in the images above.