Automated Software Testing With Macro Scheduler

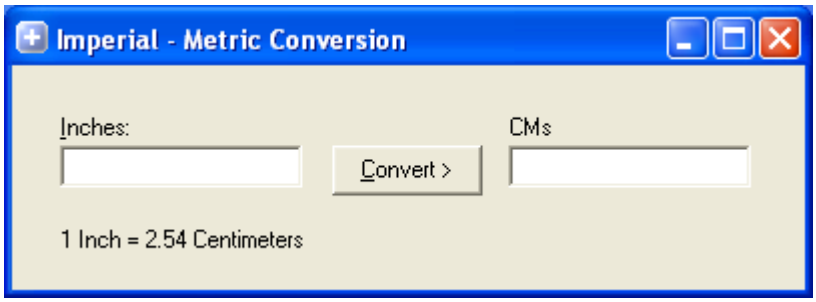
Introduction

Software testing can be a time consuming task. Traditionally QA technicians and/or programmers would sit in front of applications manually going through checklists to verify that the requirements of the software were met correctly.

Automated Testing using Macro Scheduler saves considerable time and resources. Macro Scheduler excels at testing software at the GUI level through its high level GUI functions and its ability to simulate user input.

Building a Test Script

This article will demonstrate how Macro Scheduler can be used for automated testing by way of a simple example. We will write an automated testing routine to test a small sample program, which converts inches to centimetres. Here's the example application:



Our basic test plan is this:

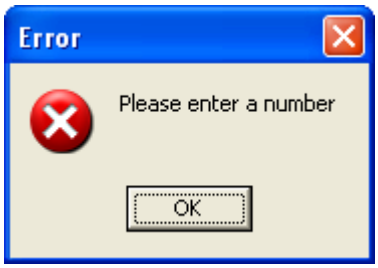
Action	Expected Outcome
Click Convert with no data entered	Should get an error box
Enter non numeric data, click Convert	Should get an error box
Enter 1 in Inches, Click Convert	Should get 2.54 in CMs box
Enter 20 in Inches, Click Convert	Should get 50.8 in CMs box

To automate this test, the first step is to click the Convert button:

```
//Focus the app
SetFocus>Imperial - Metric Conversion

//check we get an error if we click Convert without entering a number
Press ALT
Send>c
Release ALT
```

This produces the following error box with caption “Error”:



So we now want to wait for the error box to appear. We’ll wait for the error box to appear, or 5 seconds, whichever is soonest and then see if the window appeared or not:

```
//wait for error box
Let>WW_TIMEOUT=5
WaitWindowOpen>Error
If>WW_RESULT=TRUE
  //ok, the Error window appeared
  TimeStamp>logfile,No data Error: Success
  //cancel the error box
  Wait>1
  Press Esc
Else
  //No error window
  TimeStamp>logfile,No data Error: Fail
Endif
```

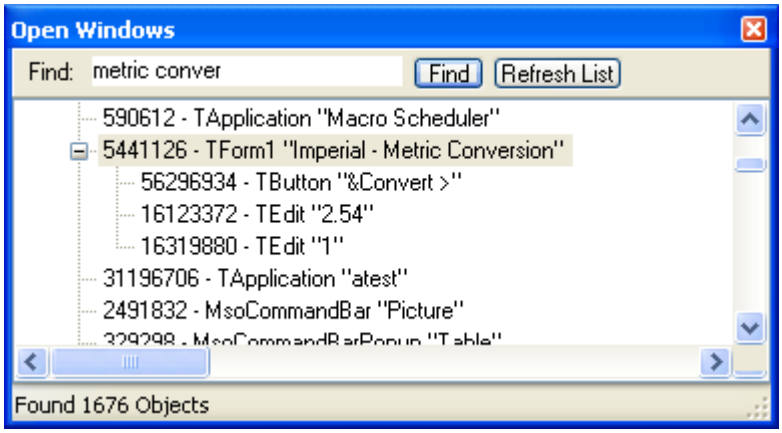
Note that we write the outcome to a log file using the TimeStamp command. This is a convenient way to write data to a log file. The TimeStamp command outputs time information so you can also see how long the process took. Alternatively you could use WriteLn, or output directly to an ODBC database, or send data to Excel using DDE, or whatever else suits your needs best.

For the second step in the test plan we use the same approach we used in the first step. The full script is presented later. For now let’s look at how we achieve the third and fourth steps where we want to get the value in the result box to see if it is what we expect. To do this we use the GetControlText function:

```
GetControlText>Imperial - Metric Conversion,TEdit,1,result
```

GetControlText takes the window title, a class name and an index and returns the text in the result variable.

Use the View System Windows Tool (from the Tools menu in Macro Scheduler) to find out the class name of objects on a window. Our sample application has two edit boxes of class TEdit.



Note that the second TEdit in the list is the Inch box and the first is the CM box. Objects are often created in reverse order to how they appear on a form. So we use index 2 to refer to the Inch edit and index 1 to refer to the CM edit.
Before pressing the Convert button we want to refocus the Inches box, clear it out and then send the data. So this whole section becomes:

```
//Send some valid data
SetFocus>Imperial - Metric Conversion

//clear inches box
SetControlText>Imperial - Metric Conversion,TEdit,2,

//focus inches box and send value (we could just use SetControlText!)
Press ALT
Send>i
Release ALT
Send>1
```

We then want to click the Convert button and extract the outcome to see if it is what we expect:

```
//Click Convert
Press ALT
Send>c
Release ALT

//result should be 2.54
GetControlText>Imperial - Metric Conversion,TEdit,1,result

If>result=2.54
  TimeStamp>logfile,1inch=2.54cm: Success
Else
  TimeStamp>logfile,1inch=2.54cm: Fail
Endif
```

We do the same thing for the fourth step in the test plan only we send 20 to the inch box instead of 1.

The Results

Let's look at the results of the test. The log file looks like this:

13:20:51:750 - No data Error: Success
13:20:52:765 - Bad data Error: Success
13:20:53:765 - 1inch=2.54cm: Success
13:20:53:781 - 20inch=50.8cm: Success

Adding Test Cases

As software products grow the need for more test cases arises. We can simply add more steps to the script, or add new test scripts and chain them together. For example, we may add another button to our sample application to reverse the conversion – to convert from centimetres to inches. We use the same methods described here but working with a different button and checking the inches field rather than the centimetres field for the outcome.

Testing Web Applications

Web pages and web applications introduce a different kind of problem. Web pages are dynamic in nature and objects do not have absolute positions. The best way to build automated test scripts for web applications is to use MacroScript WebRecorder, which builds scripts automatically by recording web activity. These scripts can then be edited to add the validation, outcome and range checks and result output needed for the test plan. For more information on WebRecorder see:
http://www.mjtnet.com/webrecorder.htm

Conclusion

Automated testing streamlines the QA process and saves precious resources, time and money. As the project continues along it's life cycle it is easy to run previously created test scripts and add new test cases. The time saved by running test scripts over manual checklists is enormous and bugs can be found more quickly. Test scenarios can easily be run repeatedly for added confidence in your product. Developers, QA engineers and customers all benefit.

With Macro Scheduler's powerful GUI automation routines, output functions, VBScript capability and complex expressions it is easy to build advanced test scenarios for all applications.

The Complete Script

```
//set up our log file here
Let>logfile=d:\testresults.log

//Focus the app
SetFocus>Imperial - Metric Conversion

//check we get an error if we click Convert without entering a number
Press ALT
Send>c
Release ALT

//wait for error box
Let>WW_TIMEOUT=5
WaitWindowOpen>Error
If>WW_RESULT=TRUE
  //ok, the Error window appeared
  TimeStamp>logfile,No data Error: Success
  //cancel the error box
  Wait>1
  Press Esc
Else
  //No error window
  TimeStamp>logfile,No data Error: Fail
Endif

//Enter some invalid data
SetFocus>Imperial - Metric Conversion
Press ALT
Send>i
Release ALT
Send>abc

//Click Convert
Press ALT
Send>c
Release ALT

//wait for error box
Let>WW_TIMEOUT=5
WaitWindowOpen>Error
If>WW_RESULT=TRUE
  //ok, the Error window appeared
  TimeStamp>logfile,Bad data Error: Success
  //cancel the error box
  Wait>1
  Press Esc
Else
  //No error window
  TimeStamp>logfile,Bad data Error: Fail
Endif

//Send some valid data
SetFocus>Imperial - Metric Conversion

//clear inches box
SetControlText>Imperial - Metric Conversion,TEdit,2,

//focus inches box and send value (we could just use SetControlText!)
Press ALT
```

```
Send>i
Release ALT
Send>1

//Click Convert
Press ALT
Send>c
Release ALT

//result should be 2.54
GetControlText>Imperial - Metric Conversion,TEdit,1,result

If>result=2.54
  TimeStamp>logfile,1inch=2.54cm: Success
Else
  TimeStamp>logfile,1inch=2.54cm: Fail
Endif

//try another value - this time we'll use SetControlText ...
SetControlText>Imperial - Metric Conversion,TEdit,2,20
Press ALT
Send>c
Release ALT
GetControlText>Imperial - Metric Conversion,TEdit,1,result
If>result=50.8
  TimeStamp>logfile,20inch=50.8cm: Success
Else
  TimeStamp>logfile,20inch=50.8cm: Fail
Endif
```