

AUTOonomyV

Samples Guide



DiJohn IC

www.dijohn-ic.com



“Making Innovation The Standard”



Table of Contents

1.0	INTRODUCTION	3
2.0	SAMPLE1_VERIFYDEFAULTS.VBS	3
3.0	SAMPLE2_VERIFYINPUTS.VBS	6
4.0	SAMPLE3_VERIFYRESET.VBS.....	11
5.0	SAMPLE4_VERIFYSUBMIT.VBS	15
6.0	SAMPLE4_VERIFYSUBMIT.VBS	19
7.0	SAMPLE5_VERIFYSTICKYMINDS.VBS	22
8.0	ALLTEST1URLSAMPLES.BAT.....	25



1.0 Introduction

This guide provides an explanation of the samples provided with the AUTOmyV tool. Samples discussed in this document include:

- Sample1_VerifyDefaults.vbs
- Sample2_VerifyInputs.vbs
- Sample3_VerifyReset.vbs
- Sample4_VerifySubmit.vbs
- Sample5_VerifyLink.vbs
- Sample6_VerifyStickminds.vbs
- AllTest1URLSamples.bat

The first 5 samples use sample web pages included in the AUTOmyV directory and require no Internet access. The 6th sample uses the www.stickyminds.com site, and does require Internet Access.

2.0 Sample1_VerifyDefaults.vbs

This sample test opens the Sample website, and checks the default values in the fields of the first screen that appears. The test has 9 main parts:

1. Comments
2. Load Library
3. Variable Declarations
4. Start Test Log Entry
5. Invoke Application
6. Get Field Data
7. Verification
8. Close Application
9. End Test Log Entry

Each is explained in the following subsections.

2.1 Comments

```
'#####  
'TEST NAME: Verify Defaults  
'  
'TEST DESCRIPTION: This test verifies all fields have the  
' correct default values.  
'#####
```

Comments such as those shown above help provide information about the test that aid in maintenance of the test script over time. Text preceded by the ' (single quote character) character is considered a comment



and is ignored by the script interpreter. Without the ' character, the line will generate an error when running the script. In this script, the Test Name and Test Description are included in the comments.

2.2 Load Library

```
'#####LOAD AUTONOMY LIBRARY#####  
Dim fso, f, file, sStrm  
file = "C:\AUTONomyV\AutonomyLib.vbs"  
Set fso = CreateObject("Scripting.FileSystemObject")  
Set f = fso.OpenTextFile(file, 1)  
sStrm = f.ReadAll  
executeGlobal(sStrm)  
'#####
```

As explained in the AUTONomyV User's Guide, these statements are required for loading the AUTONomyV library. If the AUTONomyV zip file is properly unzipped onto the C: drive, then the AutonomyLib.vbs library file will be located in the C:\AUTONomyV\ folder. For more information on FileSystemObject, OpenTextFile, and executeGlobal refer to <http://msdn.microsoft.com/>.

2.3 Variable Declarations

```
'Declare Variables  
dim f1Data  
dim f2Data  
dim f3Data  
dim f4Data  
dim f5Data  
dim f6Data
```

The `dim` statement simply declares a variable for use throughout the script. If we are going to use a value more than one, we can assign that value to a variable, and refer to the variable whenever we wish to access that value. This makes scripting simpler and more maintainable because some values may be long so it is therefore simpler to refer to a variable, which is short. Also, if the value changes, we only need to change where the value is assigned, as opposed to changing every place the value is used. This script declares six variables. These variables will be used later in the script to store information obtained from application fields.

2.4 Start Test Log Entry

```
wscript.echo "TEST START: Verify Defaults Test"  
wscript.echo "-----"
```

The `wscript.echo` statement, as described in the AUTONomyV User's Guide, is a statement that will generate a message (either in a message box or output window). These statements will simple output a message that indicates the test is about to begin.



2.5 Invoke Application

```
'Invoke URL
InvokeIE "C:\AUTOnomyV\Samples\Test1URL.html",0
```

We begin our test by invoking the application on which we plan on testing. See the AUTOnomyV User's Guide for more information on this statement.

2.6 Get Field Data

```
'Get Field Values
f1Data = Window("Mailing List form",0,0).all(WebEdit("TEXT","NAME","Last Name",0)).value
f2Data = Window("Mailing List form",0,0).all(WebEdit("TEXT","NAME","E-mail Address",0)).value
f3Data = Window("Mailing List form",0,0).all(WebCheckBox("NAME","maillist",0)).status
f4Data = Window("Mailing List form",0,0).all(WebRadioButton("NAME","mailinglistgroup",1)).status
f5Data = Window("Mailing List form",0,0).all(WebList("NAME","Priority",0)).selectedIndex
f6Data = Window("Mailing List form",0,0).all(WebTextArea("NAME","comments",0)).value
```

As previously explained, this script verifies the default values of a web page. This is done by getting the default values of the page, and comparing those values to an expected set of values. This portion of the script is responsible for getting the values from the application. The AUTOnomyV User's Guide explains these types of statements in detail, but a high level explanation is provided here.

- The first statement gets the value from a WebEdit (textbox) with the name “Last Name”, and assigns it to the f1Data variable.
- The second statement gets the value from a WebEdit (textbox) with the name “E-mail Address”, and assigns it to the f2Data variable.
- The third statement gets the Boolean value (True or False) that indicates whether or not a WebCheckBox (checkbox) with the name “maillist” is checked. The value is assigned to the f3Data variable.
- The fourth statement gets the Boolean value (True or False) that indicates whether or not the second (index of 1) WebRadioButton (radio button) in a group of radio buttons named “mailinglistgroup” is selected. The value is assigned to the f4Data variable.
- The fifth statement gets the integer index of the selected list item from a WebList (dropdown list box) with the name “Priority”, and assigns it to the f5Data variable.
- The sixth statement gets the value from a WebTextArea (text area) with the name “comments”, and assigns it to the f6Data variable.

2.7 Verification

```
'Verify Last Name Field Default Data
if f1Data <> "Enter Name" then
  wscript.echo "FAILED: The Last Name field Verification FAILED! The field contained '" & f1Data & "' but should've contained 'Enter Name'"
else
  wscript.echo "PASSED: The Last Name field Verification PASSED! The field successfully contained '" & f1Data & "'."
end if
```

Several fields are verified in this portion of the script, one of which is illustrated above. The above reveals an if-then-else statement that checks the contents of the flData variable to see if it is not (<>) equal to “Enter Name”. Since “Enter Name” is expected to be in the “Last Name” textbox, the flData variable is expected to contain the “Enter Name” text (refer to previous section). If flData is not equal to “Enter Name” an error message is generated using the wscript.echo statement. Otherwise (Else) a message is generated that indicates the step passed.

Notice that the structure of the wscript.echo statements above. The first one for example is broken up into three parts, with each of the following parts separated by the ‘&’ character:

- Part 1 → “FAIL: The Last Name field Verification FAILED! The field contained ”
- Part 2 → flData
- Part 3 → “ but should’ve contained ‘Enter Name’”

The statement is broken up like this, so that the contents of the flData variable will also be generated in the message. Characters inside of quotation marks are printed exactly as they appear, while not enclosing the flData variable in quotation marks lets the script know that this is a variable, and the value held by the variable should be printed. The ‘&’ character combines all three parts so that they may be treated as one statement.

2.8 Close Application

```
'Close Application
wscript.sleep 3000
CloseWindow "Mailing List form",0,0
```

The script closes the application at the conclusion of the test by using the CloseWindow statement (refer to the AUTOnomyV User’s Guide for more information). The ‘wscript.sleep 3000’ statement is only placed there so that someone watching the sample script run can visibly view the final statement of the screen before it is closed. 3000 is the number of milliseconds the script will wait before moving on to the next line in the script. This is of course equal to 3 seconds.

2.9 End Test Log Entry

```
wscript.echo "-----"
wscript.echo "TEST END: Verify Defaults Test"
```

These statements will output a message that indicates the test is ending.

3.0 Sample2_VerifyInputs.vbs

This sample test opens the Sample website, inputs values into the first screen that appears, and then verifies the fields hold the proper values. The test has 10 main parts:

1. Comments



2. Load Library
3. Variable Declarations
4. Start Test Log Entry
5. Invoke Application
6. Input Data Into the Form
7. Get Field Data
8. Verification
9. Close Application
10. End Test Log Entry

Each is explained in the following subsections.

3.1 Comments

```
'#####  
'TEST NAME: Verify Inputs  
'  
'TEST DESCRIPTION: This test inputs values into WebEdit  
'#####  
'#####  
'#####  
'#####  
'#####
```

Comments such as those shown above help provide information about the test that aid in maintenance of the test script over time. Text preceded by the ' (single quote character) character is considered a comment and is ignored by the script interpreter. Without the ' character, the line will generate an error when running the script. In this script, the Test Name and Test Description are included in the comments.

3.2 Load Library

```
'#####LOAD AUTONOMY LIBRARY#####  
Dim fso, f, file, sStrm  
file = "C:\AUTONOMYV\AutonomyLib.vbs"  
Set fso = CreateObject("Scripting.FileSystemObject")  
Set f = fso.OpenTextFile(file, 1)  
sStrm = f.ReadAll  
executeGlobal(sStrm)  
'#####
```

As explained in the AUTONOMYV User's Guide, these statements are required for loading the AUTONOMYV library. If the AUTONOMYV zip file is properly unzipped onto the C: drive, then the AutonomyLib.vbs library file will be located in the C:\AUTONOMYV\ folder. For more information on FileSystemObject, OpenTextFile, and executeGlobal refer to <http://msdn.microsoft.com/>.

3.3 Variable Declarations

```
'Declare Variables  
dim f1Data  
dim f2Data  
dim f3Data  
dim f4Data  
dim f5Data  
dim f6Data
```

The `dim` statement simply declares a variable for use throughout the script. If we are going to use a value more than one, we can assign that value to a variable, and refer to the variable whenever we wish to access that value. This makes scripting simpler and more maintainable because some values may be long so it is therefore simpler to refer to a variable, which is short. Also, if the value changes, we only need to change where the value is assigned, as opposed to changing every place the value is used. This script declares six variables. These variables will be used later in the script to store information obtained from application fields.

3.4 Start Test Log Entry

```
wscript.echo "TEST START: Verify Inputs Test"  
wscript.echo "-----"
```

The `wscript.echo` statement, as described in the AUTOnomyV User's Guide, is a statement that will generate a message (either in a message box or output window). These statements will simply output a message that indicates the test is about to begin.

3.5 Invoke Application

```
'Invoke URL  
InvokeIE "C:\AUTOnomyV\Samples\Test1URL.html",0
```

We begin our test by invoking the application on which we plan on testing. See the AUTOnomyV User's Guide for more information on this statement.

3.6 Input Data into Form

```
'Input data into form  
Window("Mailing List form",0,0).all(WebEdit("TEXT","NAME","Last Name",0)).value = "NewName"  
Window("Mailing List form",0,0).all(WebEdit("TEXT","NAME","E-mail Address",0)).value = "autonomyv@dijohn-ic.com"  
Window("Mailing List form",0,0).all(WebCheckBox("NAME","maillist",0)).click  
Window("Mailing List form",0,0).all(WebRadioButton("NAME","mailinglistgroup",0)).click  
Window("Mailing List form",0,0).all(WebList("NAME","Priority",0)).selectedIndex = 2  
Window("Mailing List form",0,0).all(WebTextArea("NAME","comments",0)).value = "Because I said so!"
```



This portion of the script inputs values into fields on the “Mailing List form” web page. The AUTOmyV User’s Guide explains these types of statements in detail, but a high level explanation is provided here.

- The first statement enters “NewName” into a WebEdit (textbox) with the name “Last Name”.
- The second statement enters autonomyv@dijohn-ic.com into a WebEdit (textbox) with the name “E-mail Address”.
- The third statement clicks the WebCheckButton (checkbox) with the name “maillist”, changing the state from unchecked to checked.
- The fourth statement selects the first (index of 0) WebRadioButton (radio button) in a group of radio buttons named “mailinglistgroup”.
- The fifth statement selects the third (selectedIndex = 2) item from a WebList (dropdown list box) with the name “Priority”
- The sixth statement enters “Because I said so!” into a WebTextArea (text area) with the name “comments”.

3.7 Get Field Data

```
'Get Field Values  
f1Data = Window("Mailing List form",0,0).all(WebEdit("TEXT","NAME","Last Name",0)).value  
f2Data = Window("Mailing List form",0,0).all(WebEdit("TEXT","NAME","E-mail Address",0)).value  
f3Data = Window("Mailing List form",0,0).all(WebCheckButton("NAME","maillist",0)).status  
f4Data = Window("Mailing List form",0,0).all(WebRadioButton("NAME","mailinglistgroup",0)).status  
f5Data = Window("Mailing List form",0,0).all(WebList("NAME","Priority",0)).selectedIndex  
f6Data = Window("Mailing List form",0,0).all(WebTextArea("NAME","comments",0)).value
```

As previously explained, this script verifies the values of a web page. This is done by getting the default values of the page, and comparing those values to an expected set of values. This portion of the script is responsible for getting the values from the application. The AUTOmyV User’s Guide explains these types of statements in detail, but a high level explanation is provided here.

- The first statement gets the value from a WebEdit (textbox) with the name “Last Name”, and assigns it to the f1Data variable.
- The second statement gets the value from a WebEdit (textbox) with the name “E-mail Address”, and assigns it to the f2Data variable.
- The third statement gets the Boolean value (True or False) that indicates whether or not a WebCheckButton (checkbox) with the name “maillist” is checked. The value is assigned to the f3Data variable.
- The fourth statement gets the Boolean value (True or False) that indicates whether or not the first (index of 0) WebRadioButton (radio button) in a group of radio buttons named “mailinglistgroup” is selected. The value is assigned to the f4Data variable.
- The fifth statement gets the integer index of the selected list item from a WebList (dropdown list box) with the name “Priority”, and assigns it to the f5Data variable.



- The sixth statement gets the value from a WebTextArea (text area) with the name “comments”, and assigns it to the f6Data variable.

3.8 Verification

```
'Verify Last Name Field Data
if f1Data <> "NewName" then
    wscript.echo "FAILED: The Last Name field Verification FAILED! The field contained '" & f1Data & "' but should've contained 'NewName'"
else
    wscript.echo "PASSED: The Last Name field Verification PASSED! The field successfully contained '" & f1Data & "'."
end if
```

Several fields are verified in this portion of the script, one of which is illustrated above. The above reveals an if-then-else statement that checks the contents of the f1Data variable to see if it is not (<>) equal to “NewName”. Since “NewName” is expected to be in the “Last Name” textbox, the f1Data variable is expected to contain the “NewName” text (refer to previous section). If f1Data is not equal to “NewName” an error message is generated using the wscript.echo statement. Otherwise (Else) a message is generated that indicates the step passed.

Notice that the structure of the wscript.echo statements above. The first one for example is broken up into three parts, with each of the following parts separated by the ‘&’ character:

- Part 1 → “FAIL: The Last Name field Verification FAILED! The field contained ”
- Part 2 → f1Data
- Part 3 → “ but should’ve contain ‘NewName’”

The statement is broken up like this, so that the contents of the f1Data variable will also be generated in the message. Characters inside of quotation marks are printed exactly as they appear, while not enclosing the f1Data variable in quotation marks lets the script know that this is a variable, and the value held by the variable should be printed. The ‘&’ character combines all three parts so that they may be treated as one statement.

3.9 Close Application

```
'Close Application
wscript.sleep 3000
CloseWindow "Mailing List form",0,0
```

The script closes the application at the conclusion of the test by using the CloseWindow statement (refer to the AUTOmyV User’s Guide for more information). The ‘wscript.sleep 3000’ statement is only placed there so that someone watching the sample script run can visibly view the final statement of the screen before it is closed. 3000 is the number of milliseconds the script will wait before moving on to the next line in the script. This is of course equal to 3 seconds.

3.10 End Test Log Entry

```
wscript.echo "-----"  
wscript.echo "TEST END: Verify Inputs Test"
```

These statements will output a message that indicates the test is ending.

4.0 Sample3_VerifyReset.vbs

This sample test opens the Sample website, inputs values into the first screen that appears, clicks the reset button and then verifies the fields hold the default values. The test has 11 main parts:

1. Comments
2. Load Library
3. Variable Declarations
4. Start Test Log Entry
5. Invoke Application
6. Input Data Into the Form
7. Click Reset
8. Get Field Data
9. Verification
10. Close Application
11. End Test Log Entry

Each is explained in the following subsections.

4.1 Comments

```
'#####  
'TEST NAME: Sample Verify Reset  
'  
'TEST DESCRIPTION: This test inputs values into WebEdit  
'##### fields, presses the Reset button then  
'##### verifies the fields have been  
'##### reinitialized to the defaults  
'#####
```

Comments such as those shown above help provide information about the test that aid in maintenance of the test script over time. Text preceded by the ' (single quote character) character is considered a comment and is ignored by the script interpreter. Without the ' character, the line will generate an error when running the script. In this script, the Test Name and Test Description are included in the comments.

4.2 Load Library

```
'#####LOAD AUTONOMY LIBRARY#####  
Dim fso, f, file, sStrm  
file = "C:\AUTONomyV\AutonomyLib.vbs"  
Set fso = CreateObject("Scripting.FileSystemObject")  
Set f = fso.OpenTextFile(file, 1)  
sStrm = f.ReadAll  
executeGlobal(sStrm)  
'#####
```

As explained in the AUTONomyV User's Guide, these statements are required for loading the AUTONomyV library. If the AUTONomyV zip file is properly unzipped onto the C: drive, then the AutonomyLib.vbs library file will be located in the C:\AUTONomyV\ folder. For more information on FileSystemObject, OpenTextFile, and executeGlobal refer to <http://msdn.microsoft.com/>.

4.3 Variable Declarations

```
'Declare Variables  
dim f1Data  
dim f2Data  
dim f3Data  
dim f4Data  
dim f5Data  
dim f6Data
```

The `dim` statement simply declares a variable for use throughout the script. If we are going to use a value more than one, we can assign that value to a variable, and refer to the variable whenever we wish to access that value. This makes scripting simpler and more maintainable because some values may be long so it is therefore simpler to refer to a variable, which is short. Also, if the value changes, we only need to change where the value is assigned, as opposed to changing every place the value is used. This script declares six variables. These variables will be used later in the script to store information obtained from application fields.

4.4 Start Test Log Entry

```
wscript.echo "TEST START: Verify Reset Test"  
wscript.echo "-----"
```

The `wscript.echo` statement, as described in the AUTONomyV User's Guide, is a statement that will generate a message (either in a message box or output window). These statements will simple output a message that indicates the test is about to begin.

4.5 Invoke Application



```
'Invoke URL
InvokeIE "C:\AUTOnomyV\Samples\Test1URL.html",0
```

We begin our test by invoking the application on which we plan on testing. See the AUTOnomyV User's Guide for more information on this statement.

4.6 Input Data into Form

```
'Input data into form
Window("Mailing List form",0,0).all(WebEdit("TEXT","NAME","Last Name",0)).value = "NewName"
Window("Mailing List form",0,0).all(WebEdit("TEXT","NAME","E-mail Address",0)).value = "autonomyv@dijohn-ic.com"
Window("Mailing List form",0,0).all(WebCheckButton("NAME","maillist",0)).click
Window("Mailing List form",0,0).all(WebRadioButton("NAME","mailinglistgroup",0)).click
Window("Mailing List form",0,0).all(WebList("NAME","Priority",0)).selectedIndex = 2
Window("Mailing List form",0,0).all(WebTextArea("NAME","comments",0)).value = "Because I said so!"
```

This portion of the script inputs values into fields on the “Mailing List form” web page. The AUTOnomyV User's Guide explains these types of statements in detail, but a high level explanation is provided here.

- The first statement enters “NewName” into a WebEdit (textbox) with the name “Last Name”.
- The second statement enters autonomyv@dijohn-ic.com into a WebEdit (textbox) with the name “E-mail Address”.
- The third statement clicks the WebCheckButton (checkbox) with the name “maillist”, changing the state from unchecked to checked.
- The fourth statement selects the first (index of 0) WebRadioButton (radio button) in a group of radio buttons named “mailinglistgroup”.
- The fifth statement selects the third (selectedIndex = 2) item from a WebList (dropdown list box) with the name “Priority”
- The sixth statement enters “Because I said so!” into a WebTextArea (text area) with the name “comments”.

4.7 Click Reset

```
'Click the Reset Button
Window("Mailing List form",0,0).all(WebButton("RESET","VALUE","Reset",0)).click
```

This portion of the script clicks the Reset WebButton in the “Mailing List form” page with the text “Reset” written on it (refer to AUTOnomyV User's Guide for more information).



4.8 Get Field Data

```
'Get Field Values
f1Data = Window("Mailing List form",0,0).all(WebEdit("TEXT","NAME","Last Name",0)).value
f2Data = Window("Mailing List form",0,0).all(WebEdit("TEXT","NAME","E-mail Address",0)).value
f3Data = Window("Mailing List form",0,0).all(WebCheckBox("NAME","maillist",0)).status
f4Data = Window("Mailing List form",0,0).all(WebRadioButton("NAME","mailinglistgroup",1)).status
f5Data = Window("Mailing List form",0,0).all(WebList("NAME","Priority",0)).selectedIndex
f6Data = Window("Mailing List form",0,0).all(WebTextArea("NAME","comments",0)).value
```

As previously explained, this script verifies the values of a web page. This is done by getting the default values of the page, and comparing those values to an expected set of values. This portion of the script is responsible for getting the values from the application. The AUTONomyV User's Guide explains these types of statements in detail, but a high level explanation is provided here.

- The first statement gets the value from a WebEdit (textbox) with the name “Last Name”, and assigns it to the f1Data variable.
- The second statement gets the value from a WebEdit (textbox) with the name “E-mail Address”, and assigns it to the f2Data variable.
- The third statement gets the Boolean value (True or False) that indicates whether or not a WebCheckBox (checkbox) with the name “maillist” is checked. The value is assigned to the f3Data variable.
- The fourth statement gets the Boolean value (True or False) that indicates whether or not the second (index of 1) WebRadioButton (radio button) in a group of radio buttons named “mailinglistgroup” is selected. The value is assigned to the f4Data variable.
- The fifth statement gets the integer index of the selected list item from a WebList (dropdown list box) with the name “Priority”, and assigns it to the f5Data variable.
- The sixth statement gets the value from a WebTextArea (text area) with the name “comments”, and assigns it to the f6Data variable.

4.9 Verification

```
'Verify Last Name Field Default Data
if f1Data <> "Enter Name" then
  wscript.echo "FAILED: The Last Name field Verification FAILED! The field contained '" & f1Data & "' but should've contained 'Enter Name'"
else
  wscript.echo "PASSED: The Last Name field Verification PASSED! The field successfully contained '" & f1Data & "'."
end if
```

Several fields are verified in this portion of the script, one of which is illustrated above. The above reveals an if-then-else statement that checks the contents of the f1Data variable to see if it is not (<>) equal to “Enter Name”. Since “Enter Name” is expected to be in the “Last Name” textbox, the f1Data variable is expected to contain the “Enter Name” text (refer to previous section). If f1Data is not equal to “Enter Name” an error message is generated using the wscript.echo statement. Otherwise (Else) a message is generated that indicates the step passed.

Notice that the structure of the wscript.echo statements above. The first one for example is broken up into three parts, with each of the following parts separated by the ‘&’ character:

- Part 1 → “FAIL: The Last Name field Verification FAILED! The field contained ”
- Part 2 → f1Data
- Part 3 → “ but should’ve contain ‘ Enter Name’”

The statement is broken up like this, so that the contents of the f1Data variable will also be generated in the message. Characters inside of quotation marks are printed exactly as they appear, while not enclosing the f1Data variable in quotation marks lets the script know that this is a variable, and the value held by the variable should be printed. The ‘&’ character combines all three parts so that they may be treated as one statement.

4.10 Close Application

```
'Close Application  
wscript.sleep 3000  
CloseWindow "Mailing List form",0,0
```

The script closes the application at the conclusion of the test by using the CloseWindow statement (refer to the AUTonomyV User’s Guide for more information). The ‘wscript.sleep 3000’ statement is only placed there so that someone watching the sample script run can visibly view the final statement of the screen before it is closed. 3000 is the number of milliseconds the script will wait before moving on to the next line in the script. This is of course equal to 3 seconds.

4.11 End Test Log Entry

```
wscript.echo "-----"  
wscript.echo "TEST END: Verify Reset Test"
```

These statements will output a message that indicates the test is ending.

5.0 Sample4_VerifySubmit.vbs

This sample test opens the Sample website, inputs values into the first screen that appears, clicks the Send Request button and then verifies some text on the resulting screen. The test has 10 main parts:

1. Comments
2. Load Library
3. Variable Declarations
4. Start Test Log Entry
5. Invoke Application
6. Input Data Into the Form
7. Click Send Request
8. Verification
9. Close Application

10. End Test Log Entry

Each is explained in the following subsections.

5.1 Comments

```
#####  
'TEST NAME: Verify Submit  
'  
'TEST DESCRIPTION: This test inputs values into WebEdit  
' fields, presses the Send Request button  
' then verifies part of message on the new  
' screen that comes up.  
#####
```

Comments such as those shown above help provide information about the test that aid in maintenance of the test script over time. Text preceded by the ' (single quote character) character is considered a comment and is ignored by the script interpreter. Without the ' character, the line will generate an error when running the script. In this script, the Test Name and Test Description are included in the comments.

5.2 Load Library

```
#####LOAD AUTONOMY LIBRARY#####  
Dim fso, f, file, sStrm  
file = "C:\AUTOmomyV\AutonomyLib.vbs"  
Set fso = CreateObject("Scripting.FileSystemObject")  
Set f = fso.OpenTextFile(file, 1)  
sStrm = f.ReadAll  
executeGlobal(sStrm)  
#####
```

As explained in the AUTOmomyV User's Guide, these statements are required for loading the AUTOmomyV library. If the AUTOmomyV zip file is properly unzipped onto the C: drive, then the AutonomyLib.vbs library file will be located in the C:\AUTOmomyV\ folder. For more information on FileSystemObject, OpenTextFile, and executeGlobal refer to <http://msdn.microsoft.com/>.

5.3 Variable Declarations

```
'Declare Variables  
dim txtRng  
dim txtFnd
```

The `dim` statement simply declares a variable for use throughout the script. If we are going to use a value more than one, we can assign that value to a variable, and refer to the variable whenever we wish to access that value. This makes scripting simpler and more maintainable because some values may be long so it is therefore simpler to refer to a variable, which is short. Also, if the value changes, we only need to



change where the value is assigned, as opposed to changing every place the value is used. This script declares two variables. The first variable will be used later in the script to store the existing screen text, while the second will be used later in the script to store a Boolean value (True or False) indicating whether or not a search string was found within the screen text.

5.4 Start Test Log Entry

```
wscript.echo "TEST START: Verify Submit Test"  
wscript.echo "-----"
```

The wscript.echo statement, as described in the AUTOmomyV User's Guide, is a statement that will generate a message (either in a message box or output window). These statements will simple output a message that indicates the test is about to begin.

5.5 Invoke Application

```
'Invoke URL  
InvokeIE "C:\AUTOmomyV\Samples\Test1URL.html",0
```

We begin our test by invoking the application on which we plan on testing. See the AUTOmomyV User's Guide for more information on this statement.

5.6 Input Data into Form

```
'Input data into form  
Window("Mailing List form",0,0).all(WebEdit("TEXT","NAME","Last Name",0)).value = "NewName"  
Window("Mailing List form",0,0).all(WebEdit("TEXT","NAME","E-mail Address",0)).value = "autonomyv@dijohn-ic.com"  
Window("Mailing List form",0,0).all(WebCheckBox("NAME","maillist",0)).click  
Window("Mailing List form",0,0).all(WebRadioButton("NAME","mailinglistgroup",0)).click  
Window("Mailing List form",0,0).all(WebList("NAME","Priority",0)).selectedIndex = 2  
Window("Mailing List form",0,0).all(WebTextArea("NAME","comments",0)).value = "Because I said so!"
```

This portion of the script inputs values into fields on the “Mailing List form” web page. The AUTOmomyV User's Guide explains these types of statements in detail, but a high level explanation is provided here.

- The first statement enters “NewName” into a WebEdit (textbox) with the name “Last Name”.
- The second statement enters autonomyv@dijohn-ic.com into a WebEdit (textbox) with the name “E-mail Address”.
- The third statement clicks the WebCheckBox (checkbox) with the name “maillist”, changing the state from unchecked to checked.
- The fourth statement selects the first (index of 0) WebRadioButton (radio button) in a group of radio buttons named “mailinglistgroup”.



- The fifth statement selects the third (selectedIndex = 2) item from a WebList (dropdown list box) with the name “Priority”
- The sixth statement enters “Because I said so!” into a WebTextArea (text area) with the name “comments”.

5.7 Click Send Request

```
'Click the Send Request button  
Window("Mailing List form",0,0).all(WebButton("SUBMIT","VALUE","Send Request",0)).click
```

This portion of the script clicks the Send Request WebButton in the “Mailing List form” page with the text “Send Request” written on it (refer to AUTOnomyV User’s Guide for more information).

5.8 Verification

```
'Check for Congratulations on the new 'Test Successful' screen that appears  
set txtRng = Window("Test Successful",0,0).body.createTextRange  
txtFnd = txtRng.findText("Congratulations")  
  
if txtFnd <> True then  
    wscript.echo "FAILED: The word 'Congratulations' was not found."  
else  
    wscript.echo "PASSED: The word 'Congratulations' was successfully found."  
end if
```

This portion of the script is designed to verify that the word “Congratulations” appears on the screen that results from clicking the Send Request button. This is accomplished by first capturing all of the text that exists on the screen, and assigning it to the txtRng variable in the second statement shown above. Performing the createTextRange method on the body of the HTML page is crucial to doing this (see <http://msdn.microsoft.com> for more information on createTextRange). The next statement determines whether or not the word “Congratulations” exists in the screen text, and assigns the True/False answer to the txtFnd variable. An if-then-else statement is then used to check the contents of the txtFnd variable to see if it is not (<>) equal to “True”. Since “True” is expected to be in the txtFnd variable (because “Congratulations” is expected to be on the page), an error message is generated using the wscript.echo statement if it is not “True”. Otherwise (Else) a message is generated that indicates the step passed.

5.9 Close Application

```
'Close Application  
wscript.sleep 3000  
CloseWindow "Test Successful",0,0
```

The script closes the application at the conclusion of the test by using the CloseWindow statement (refer to the AUTOmyV User's Guide for more information). The 'wscript.sleep 3000' statement is only placed there so that someone watching the sample script run can visibly view the final statement of the screen before it is closed. 3000 is the number of milliseconds the script will wait before moving on to the next line in the script. This is of course equal to 3 seconds.

5.10 End Test Log Entry

```
wscript.echo "-----"  
wscript.echo "TEST END: Verify Submit Test"
```

These statements will output a message that indicates the test is ending.

6.0 Sample4_VerifySubmit.vbs

This sample test opens the Sample website, clicks the "Link Page" link and then verifies some text on the resulting screen. The test has 9 main parts:

1. Comments
2. Load Library
3. Variable Declarations
4. Start Test Log Entry
5. Invoke Application
6. Click "Link Page" link
7. Verification
8. Close Application
9. End Test Log Entry

Each is explained in the following subsections.

6.1 Comments

```
'#####  
'TEST NAME: Verify Link  
'  
'TEST DESCRIPTION: This test clicks the link and then  
'##### verifies part of the message on the new  
'##### screen that comes up. Then closes the window.  
'#####
```

Comments such as those shown above help provide information about the test that aid in maintenance of the test script over time. Text preceded by the ' (single quote character) character is considered a comment and is ignored by the script interpreter. Without the ' character, the line will generate an error when running the script. In this script, the Test Name and Test Description are included in the comments.

6.2 Load Library

```
'#####LOAD AUTONOMY LIBRARY#####'  
Dim fso, f, file, sStrm  
file = "C:\AUTONomyV\AutonomyLib.vbs"  
Set fso = CreateObject("Scripting.FileSystemObject")  
Set f = fso.OpenTextFile(file, 1)  
sStrm = f.ReadAll  
executeGlobal(sStrm)  
'#####'
```

As explained in the AUTONomyV User's Guide, these statements are required for loading the AUTONomyV library. If the AUTONomyV zip file is properly unzipped onto the C: drive, then the AutonomyLib.vbs library file will be located in the C:\AUTONomyV\ folder. For more information on FileSystemObject, OpenTextFile, and executeGlobal refer to <http://msdn.microsoft.com/>.

6.3 Variable Declarations

```
'Declare Variables  
dim txtRng  
dim txtFnd
```

The `dim` statement simply declares a variable for use throughout the script. If we are going to use a value more than one, we can assign that value to a variable, and refer to the variable whenever we wish to access that value. This makes scripting simpler and more maintainable because some values may be long so it is therefore simpler to refer to a variable, which is short. Also, if the value changes, we only need to change where the value is assigned, as opposed to changing every place the value is used. This script declares two variables. The first variable will be used later in the script to store the existing screen text, while the second will be used later in the script to store a Boolean value (True or False) indicating whether or not a search string was found within the screen text.

6.4 Start Test Log Entry

```
wscript.echo "TEST START: Verify Link Test"  
wscript.echo "-----"
```

The `wscript.echo` statement, as described in the AUTONomyV User's Guide, is a statement that will generate a message (either in a message box or output window). These statements will simple output a message that indicates the test is about to begin.

6.5 Invoke Application



```
'Invoke URL  
InvokeIE "C:\AUTOnomyV\Samples\Test1URL.html",0
```

We begin our test by invoking the application on which we plan on testing. See the AUTOnomyV User's Guide for more information on this statement.

6.6 Click Link Page

```
'Click the 'Link Page' link  
Window("Mailing List form",0,0).all(WebLink("INNERTEXT","Link Page",0)).click
```

This portion of the script clicks the link in the “Mailing List form” page with the text “Link Page” (refer to AUTOnomyV User's Guide for more information).

6.7 Verification

```
'Check for Congratulations on the new 'Test Successful' screen that appears  
set txtRng = Window("Link Page",0,0).body.createTextRange  
txtFnd = txtRng.findText("the Link Screen")  
  
if txtFnd <> True then  
    wscript.echo "FAILED: The text 'the Link Screen' was not found."  
else  
    wscript.echo "PASSED: The text 'the Link Screen' was successfully found."  
end if
```

This portion of the script is designed to verify that the words “the Link Screen” appears on the screen that results from clicking the “Link Page” link. This is accomplished by first capturing all of the text that exists on the screen, and assigning it to the txtRng variable in the first statement shown above. Performing the createTextRange method on the body of the HTML page is crucial to doing this (see <http://msdn.microsoft.com> for more information on createTextRange). The next statement determines whether or not the words “the Link Screen” exists in the screen text, and assigns the True/False answer to the txtFnd variable. An if-then-else statement is then used to check the contents of the txtFnd variable to see if it is not (<>) equal to “True”. Since “True” is expected to be in the txtFnd variable (because “the Link Screen” is expected to be on the page), an error message is generated using the wscript.echo statement if it is not “True”. Otherwise (Else) a message is generated that indicates the step passed.

6.8 Close Application

```
'Close Application  
wscript.sleep 3000  
CloseWindow "Link Page",0,0
```

The script closes the application at the conclusion of the test by using the CloseWindow statement (refer to the AUTOnomyV User's Guide for more information). The 'wscript.sleep 3000' statement is only placed there so that someone watching the sample script run can visibly view the final statement of the screen before it is closed. 3000 is the number of milliseconds the script will wait before moving on to the next line in the script. This is of course equal to 3 seconds.

6.9 End Test Log Entry

```
wscript.echo "-----"  
wscript.echo "TEST END: Verify Link Test"
```

These statements will output a message that indicates the test is ending.

7.0 Sample5_VerifyStickyMinds.vbs

This sample test opens the www.stickyminds.com website, enters a search string, executes the search and then verifies some text on the resulting screen. The test requires an Internet connection and has 10 main parts:

1. Comments
2. Load Library
3. Variable Declarations
4. Start Test Log Entry
5. Invoke Application
6. Input a Value into Search Field
7. Click "Go" Button
8. Verification
9. Close Application
10. End Test Log Entry

Each is explained in the following subsections.

7.1 Comments

```
'#####  
'TEST NAME: Verify StickyMinds Search  
'  
'TEST DESCRIPTION: This test performs a search in the  
'##### www.stickyminds.com website and verifies  
'##### text on the resulting page.  
'#####
```

Comments such as those shown above help provide information about the test that aid in maintenance of the test script over time. Text preceded by the ' (single quote character) character is considered a comment

and is ignored by the script interpreter. Without the ' character, the line will generate an error when running the script. In this script, the Test Name and Test Description are included in the comments.

7.2 Load Library

```
'#####LOAD AUTONOMY LIBRARY#####  
Dim fso, f, file, sStrm  
file = "C:\AUTONomyV\AutonomyLib.vbs"  
Set fso = CreateObject("Scripting.FileSystemObject")  
Set f = fso.OpenTextFile(file, 1)  
sStrm = f.ReadAll  
executeGlobal(sStrm)  
'#####
```

As explained in the AUTONomyV User's Guide, these statements are required for loading the AUTONomyV library. If the AUTONomyV zip file is properly unzipped onto the C: drive, then the AutonomyLib.vbs library file will be located in the C:\AUTONomyV\ folder. For more information on FileSystemObject, OpenTextFile, and executeGlobal refer to <http://msdn.microsoft.com/>.

7.3 Variable Declarations

```
'Declare Variables  
dim txtRng  
dim txtFnd
```

The `dim` statement simply declares a variable for use throughout the script. If we are going to use a value more than one, we can assign that value to a variable, and refer to the variable whenever we wish to access that value. This makes scripting simpler and more maintainable because some values may be long so it is therefore simpler to refer to a variable, which is short. Also, if the value changes, we only need to change where the value is assigned, as opposed to changing every place the value is used. This script declares two variables. The first variable will be used later in the script to store the existing screen text, while the second will be used later in the script to store a Boolean value (True or False) indicating whether or not a search string was found within the screen text.

7.4 Start Test Log Entry

```
wscript.echo "TEST START: Verify Stickyminds Search Test"  
wscript.echo "-----"
```

The `wscript.echo` statement, as described in the AUTONomyV User's Guide, is a statement that will generate a message (either in a message box or output window). These statements will simple output a message that indicates the test is about to begin.

7.5 Invoke Application

```
'Invoke the application
InvokeIE "www.stickyminds.com",10
```

We begin our test by invoking the application on which we plan on testing. See the AUTOnomyV User's Guide for more information on this statement.

7.6 Input a Value into Search Field

```
'Input a value into the search field
Window("StickyMinds Home Page",0,10).all(WebEdit("TEXT","NAME","freetext",0)).value = "dion johnson"
```

This portion of the script inputs a value into a field on the “StickyMinds Home Page” web page. The AUTOnomyV User's Guide explains these types of statements in detail, but a high level explanation is provided here.

- The first statement waits up-to 10 seconds for the “StickyMinds Home Page” web page to finish loading, then enters “dion johnson” into a WebEdit (textbox) with the name “freetext”.

7.7 Click Go Button

```
'Click Go Button
Window("StickyMinds Home Page",0,10).all(WebImage("NAME","submit",0)).click
```

This portion of the script clicks the image button in the “StickyMinds Home Page” page with the name “submit” (refer to AUTOnomyV User's Guide for more information).

7.8 Verification

```
'Check for text
set txtRng = Window("StickyMinds.com : Site Search",0,10).body.createTextRange
txtFnd = txtRng.findText("Your request for topic dion johnson")

if txtFnd <> True then
    wscript.echo "FAILED: The text 'Your request for topic dion johnson' was not found."
else
    wscript.echo "PASSED: The text 'Your request for topic dion johnson' was successfully found."
end if
```

This portion of the script is designed to verify that the words “Your request for topic dion johnson” appears on the screen that results from clicking the “Go” button. This is accomplished by first capturing all of the text that exists on the screen, and assigning it to the txtRng variable in the first statement shown above. Performing the createTextRange method on the body of the HTML page is crucial to doing this (see <http://msdn.microsoft.com> for more information on createTextRange). The next statement determines whether or not the words “the Link Screen” exists in the screen text, and assigns the True/False answer to the txtFnd variable. An if-then-else statement is then used to check the contents of the txtFnd variable to see if it is not (<>) equal to “True”. Since “True” is expected to be in the txtFnd variable (because “Your



request for topic dion johnson” is expected to be on the page), an error message is generated using the wscript.echo statement if it is not “True”. Otherwise (Else) a message is generated that indicates the step passed.

7.9 Close Application

```
'Close Application  
wscript.sleep 3000  
CloseWindow "StickyMinds.com : Site Search",0,0
```

The script closes the application at the conclusion of the test by using the CloseWindow statement (refer to the AUTOnomyV User’s Guide for more information). The ‘wscript.sleep 3000’ statement is only placed there so that someone watching the sample script run can visibly view the final statement of the screen before it is closed. 3000 is the number of milliseconds the script will wait before moving on to the next line in the script. This is of course equal to 3 seconds.

7.10 End Test Log Entry

```
wscript.echo "-----"  
wscript.echo "TEST END: Verify Stickyminds Search Test"
```

These statements will output a message that indicates the test is ending.

8.0 AllTest1URLSamples.bat

This is simply a windows batch file that executes Samples 1 through 5 in succession as if executing from the command line. There are two main parts to the script:

- Comments
- Script Calls

Each part is explained in the following subsections.

8.1 Comments

```
REM Execute all samples that are performed on the Test1URL.html application
```

Comments such as the one shown above help provide information about the test that aid in maintenance of the test script over time. Text preceded by “REM” (short of remark) statement is considered a comment and is ignored by the script interpreter. In addition, the statements will be printed to the output screen. Without the “REM” statement, the line will generate an error when running the script.

8.2 Script Calls



```
cscript c:\AUTOnomyV\Samples\Sample1_VerifyDefaults.vbs  
cscript c:\AUTOnomyV\Samples\Sample2_VerifyInputs.vbs  
cscript c:\AUTOnomyV\Samples\Sample3_VerifyReset.vbs  
cscript c:\AUTOnomyV\Samples\Sample4_VerifySubmit.vbs  
cscript c:\AUTOnomyV\Samples\Sample5_VerifyLink.vbs
```

Each statement above executes a script. Running this .bat file is equivalent to typing in each one of these lines individually from the command line.