

Guy's Top 10 Ezines

Contents (In order of popularity)

- [1\) Ezine 26 - Run Send Keys and Sleep](#)
- [2\) Ezine 23 - Enable Active Directory Accounts](#)
- [3\) Ezine 16 - Printer Scripts](#)
- [4\) Ezine 68 - Stop Start Services](#)
- [5\) Ezine 36 - FSO File System Object](#)
- [6\) Ezine 28 - InStr](#)
- [7\) Ezine 46 - MsgBox\(\)](#)
- [8\) Ezine 18 - Underscore Syntax](#)
- [9\) Ezine 50- pwdLastSet](#)
- [10\) Ezine 45 - Shutdown](#)

To get the most out of my scripts, you need a top class VBScript Editor, [try a free download of OnScript](#)

Contents Ezine 26 - Run, SendKeys and Sleep

1. This week's secret
2. Example 1 - Creating a VBscript which will open Notepad
3. Guy's Out Takes - Answer

This week s secret - Everything has a purpose

I will let you into a training secret. When I run courses, delegates prefer me to take a view about a topic rather than sit on the fence. For instance, I once said that WScript.Sleep was useless - why would you ever want a script to wait? Well, when I take a view someone will take a contrary position and we get a good discussion.

In truth I sometimes change my mind, especially when I remember this: everything has a use under the right circumstances. So in the case of WScript.Sleep, I have made an about turn, I now realise that on occasions script runs faster than the program that they are controlling. By employing the command WScript.Sleep you can make the script wait until the program is ready for the next instruction. The benefit is no error message, and the script achieves its purpose.

Example 1 - Creating a VBscript to open Notepad

This week is your chance to have a bit of fun, to just learn by messing about with scripts. My point is that this example of opening Notepad.exe is not a crucial script, but it does give valuable practice with the .run method.

My script works at two levels. At the practical level, VBScript will open, run, or execute a program. This is three ways of saying, VBScript can start executables like Notepad, WinWord or even IExplore. At the theoretical level, you will learn about the WshShell.Run method; and also discover when to pause scripts with Wscript.Sleep command. As a bonus, you will have a chance to manipulate text within your program thanks to WshShell.SendKeys.

Warning.

As you may have spotted, I am not a great one for long disclaimers, so please take note of my warning about SendKeys. Most of the time SendKeys works as expected. However, if you had the proverbial barrow load of monkeys, all using SendKeys, then one of them would do something drastic like over-write a vital report that just happen to be open. My advice is: take sensible precautions, close programs that you are not using before experimenting with SendKeys.

Instructions for opening Notepad

1. Copy and paste the script below into notepad.
2. Save the file with .vbs extension e.g. SendKeys.vbs.
3. Double click and observe Notepad open.
4. Be amazed to see a phantom typist.

```
' SendKeys.vbs
' Example VBScript Run Notepad and use SendKeys
' Author Guy Thomas http://computerperformance.co.uk/
' Ezine 26 Version 1.8 - June 2005
' -----'
Option Explicit
Dim objShell, Racey, intCount
Set objShell = CreateObject("WScript.Shell")
objShell.Run "notepad"
Wscript.Sleep 1500
Racey = 1000
intCount=0

Do While intCount < 7
objShell.SendKeys "Hello Guy"
objShell.SendKeys "{TAB}"
objShell.SendKeys "This is line: " & intCount
objShell.SendKeys "{ENTER}"
WScript.Sleep Racey
intCount = intCount + 1
Racey = Racey - 100
Loop
objShell.SendKeys "%F"
WScript.Sleep 1500
objShell.SendKeys "x"
WScript.Sleep 1500
objShell.SendKeys "{TAB}"
WScript.Sleep 500
objShell.SendKeys "{ENTER}"

WScript.Quit
' End of Example SendKeys VBScript
```

Learning Points

Note 1: Line 9 Wscript.Sleep 1500. This command makes sure that the script waits for notepad to open before continuing with the SendKeys commands.

Note 2: Spot the { } braces for special characters, for example carriage return or tab.

Note 3: I challenge you to experiment with the command, 'Do While... loop. If you want a bigger challenge substitute, For Next.

Note 4: Can you see what variable Racey is doing? Did you notice that each line is written progressively faster?

Challenge:

Try other programs like Winword in place of Notepad. It's also fun to open web pages with objShell.Run "http:xxxx". However, before you try this, remember my warning about SendKeys. To begin with, remove all lines after the .Run command, which contain SendKeys.

This is an example of .send method, but with the HTTP variation of objShell.Run: "http://computerperformance.co.uk/"

```
' Web.vbs
' Example VBScript Run Notepad and use SendKeys
' Author Guy Thomas http://computerperformance.co.uk/
' Version 1.2 - April 18th 2004
' -----'
Option Explicit
Dim objShell
Set objShell = CreateObject("WScript.Shell")
objShell.Run "http://computerperformance.co.uk/"
WScript.Quit

' End of Example VBScript .run method
```

Summary:

This Example uses the WshShell.Run method. In addition the instructions are designed to give you experience of Sleep, and SendKeys. All the while, you will be re-enforcing other VBScript techniques, for example looping.

Contents for Ezine 23 - Enabling User Accounts

1. This week's secret
2. Script to Enable accounts that have been 'Disabled' X
3. Force users to change password at next logon
4. Out Takes and Answers
5. See more on UserAccountControl

Please note:

To force users to change password at next logon you need both these properties, UserAccountControl and pwdLastSet. Since writing the initial Ezine 23, I have discovered the importance of adding: pwdLastSet to this example script.

This week s secret - CSVDE does not set passwords

CSVDE is a wonderful utility for importing users from a spreadsheet into Active Directory. Unfortunately, you cannot use CSVDE to set passwords. This has repercussions where you have password Group Policies. In such cases zero length passwords are not permitted.

To spell out the problem: if your domain account policy means that passwords have to be a minimum of 6 characters, then you cannot import accounts with CSVDE and set them to 'Enabled'. Nor can you set them to: 'User must change password at next logon'. All that you can do is import the user accounts with CSVDE and create a VBScript to add the passwords and to enable the accounts.

In fact, if you attempt to add a password field to your CSVDE spreadsheet, then import fails with an unfriendly error message. Worse, it seems whenever I try to add a password field to a CSVDE import, the operating system gets so upset by this illegal procedure that I have to start again with a new spreadsheet.

The answer is a VBScript to set the UserAccountControl (and add pwdLastSet).

A Script to Enable accounts that have been 'Disabled' X

The purpose of this script is to enable accounts so that users can logon to your domain. The situation is that you have just bulk imported users but all the accounts are disabled. You want people to be able to use their new accounts.

The key LDAP property is UserAccountControl, what we need to do is change the value from 514 to 512. With a value of 512, the account will be enabled and the users can logon successfully.

Instructions to Enable Active Directory User Accounts

1. Pre-requisites. You need either a Windows 2000 or Server 2003 domain controller for this script to work.
2. Change Line 11 "OU=Cowbridge ," to the name of one of your OUs. Alternatively, create an OU called Cowbridge.
3. Please make sure that the OU has users and that their accounts are disabled. (Right Click any account, select ('Disable Account')
4. Copy and paste the script below into notepad.
5. Save the file with .vbs extension e.g. AccountControl.vbs
6. Double click and observe the message box
7. Importance of adding : pwdLastSet

```
' Set AccountControl.vbs
' Example VBScript to enable user accounts in a named OU
' Author Guy Thomas http://computerperformance.co.uk/
' Version 1.7 - March 21st 2004
' -----'
Option Explicit
Dim objOU, objUser, objRootDSE
Dim strContainer, strLastUser, strDNSDomain, intCounter, intAccValue
Set objRootDSE = GetObject("LDAP://RootDSE")
strDNSDomain = objRootDSE.Get("DefaultNamingContext")
strContainer = "OU=Cowbridge ,"
intAccValue = 512
strContainer = strContainer & strDNSDomain
set objOU =GetObject("LDAP://" & strContainer )
intCounter = 0
  For each objUser in objOU
    If objUser.class="user" then
      objUser.Put "userAccountControl", intAccValue
      objUser.SetInfo
      intCounter = intCounter +1
      strLastUser = objUser.Get ("name")
    End if
  next
WScript.Echo intCounter & " Accounts Enabled. Value " _
& intAccValue
WScript.Quit

' End of VBScript Example
```

Learning Points

Note 1: The key method here is `.Put`, for example `objUser.Put`

Note 2: If `objUser.class = "user"` Here we only wish to enable user accounts not computer accounts.

Note 3: `intAccValue` allows me to echo the value that I have set for `UserAccountControl`. This is useful if I wish to experiment with the values below.

Troubleshooting

Naturally you have to create or move some accounts into the OU = Cowbridge and make sure they are disabled.

If you are testing a scripts for the second or third time, then you need to Refresh to check that the amendments are working. The secret is to select the OU, then choose Refresh, from the short cut menu. For some strange reason F5 (Function key 5) only works the first time.

UserAccountControl Values

You may be wondering what range of settings you can use on the `UserAccountControl` attribute. Here is a list of the most common values for a user object.

512 - Enable Account

514 - Disable account

544 - Account Enabled - Require user to change password at first logon

66048 - Password never expires

262656 - Smart Card Logon Required

I discovered the above values by experimenting with the users' property sheets in Active Directory Users and Computers. What I did was set the check boxes in the Account property tab and then exported the users with `CSVDE -f account.csv`. In truth, I used the `-d` switch to filter the records so that I only exported users in the Cowbridge OU.

```
CSVDE -f account.csv -d "ou=cowbridge,dc=cp,dc=com"
```

Finally, I examined the `UserAccountControl` column in the spreadsheet, and compared the values with ticks in checkboxes under the Account tab.

Force users to change password at next logon

This is where we put it all together. A combination of last week's script to set user accounts password, with this week's script to enable the accounts. Because I want them to change password at next logon, I set the UserAccountControl to be 544.

```
' Set AccPwd.vbs
' VBScript to require users change passwords at next logon
' Author Guy Thomas http://computerperformance.co.uk/
' Version 2.2 - March 21st 2004
' -----'
Option Explicit
Dim objOU, objUser, objRootDSE
Dim strContainer, strLastUser, strDNSDomain, intCounter, intAccValue
Set objRootDSE = GetObject("LDAP://RootDSE")
strDNSDomain = objRootDSE.Get("DefaultNamingContext")
strContainer = "OU=Cowbridge ,"
intAccValue = 544
strContainer = strContainer & strDNSDomain
set objOU =GetObject("LDAP://" & strContainer )
intCounter = 0
  For each objUser in objOU
    If objUser.class="user" then
      objUser.SetPassword "P@$$er2004"
      objUser.SetInfo
      objUser.Put "userAccountControl", intAccValue
      objUser.SetInfo
      intCounter = intCounter + 1
      strLastUser = objUser.Get ("name")
    End if
  next
WScript.Echo intCounter & " Users change pwd next logon. Value " _
& intAccValue
WScript.Quit
```

Learning Points

Note 1: intAccValue is now changed to 544.

Note 2: We insert last week's method, objUser.SetPassword.

Contents of Ezine No 16 - Printer Scripts

1. This week's secret
2. Example 1: Mapping your printer
3. Example 2: Deleting a printer
4. Example 3: Mapping multiple printers
5. Map a Printer to a Computer: RUNDLL32 PRINTUI.DLL,PrintUIEntry
6. Summary

This week's secret

If something seems reasonable, but then I am told that it cannot be done, its like a red rag to a bull. For example, I was told that you cannot assign a printer to a COMPUTER with a logon script. Now I have a good basic VBScript that creates a printer, and it works fine when I double click the .vbs file. If I assign my printer script to a user - no problem. However, when I assign it to a computer as a Start Up script, it will not work. Just to check, I created a non-printer script and assigned it to Start up and that does work.

Postscript: John G wrote in with information on RUNDLL32 PrintUI.DLL. **Here is what I found:** <http://computerperformance.co.uk/ezine/ezine16a.htm>

Example 1: Mapping your Printer

Scenario: Your network has a printer device on a network server. Your users would like to send their documents to this printer and your task is to assign this network printer as easily as possible.

Instructions for Mapping a Printer

Change "\\YourServer\PrinterName" to reflect a real server and a real printer on your network, otherwise you will get an 800 WSH error message.

Copy the following script. Next open Notepad, now paste in the text, when you have altered the UNCpath, save the file with a .vbs extension, for example: Printer.vbs.

```
' Example VBScript to map a printer on a server
' Guy Thomas February 2004.
' *****
Option Explicit
Dim netPrinter, UNCpath
UNCpath = "\\YourServer\PrinterName"
Set netPrinter = CreateObject("WScript.Network")
netPrinter.AddWindowsPrinterConnection UNCpath
WScript.Echo "Your printer is mapped from : " & UNCpath
WScript.Quit
' End of Script
```

Note 1: The UNC path, "\\ServerName\PrinterName", to your printer is enclosed in speech marks. (You have changed the names haven't you?)

Note 2: Line 7 Creates a network object called netPrinter. As netPrinter a variable, you could change the name to HP6L if you wished.

Note 3: On line 8 AddWindowsPrinterConnection is a Method. Unlike variables, you cannot just make up names of methods.

Note 4: Line 9 adds a WScript.Echo which confirms the path of the mapped printer.

Error Codes generated by this printer logon scripts

80070043 The network name cannot be found (Mistyped Server, or Printer name)

80070055 The local device name is already in use (Already created a printer)

800704B3 No network provider accepted the given network path

80070709 The Printer name is invalid

80071329 Object Already Exists

80072030 There is no such object on the server

8007203A The server is not operational

Example 2 Delete a Printer with - RemovePrinterConnection

I recommend the RemovePrinterConnection method when testing your printer scripts. I challenge you to create a pair of scripts, one script which creates a network printer, and another to remove it. Always be on the lookout for such 'mirror image' scripts when testing and learning about VBScripting.

Instructions.

Copy and paste into notepad, save with .vbs extension.

As with Example 1, change the UNCpath.

```
' Example VBScript to remove a Printer
' Guy Thomas February 2004.
' *****
Option Explicit
Dim DelPrint, UNCpath
UNCpath = "\\YourServer\PrinterName"
Set DelPrint = WScript.CreateObject("WScript.Network")
DelPrint.RemovePrinterConnection UNCpath, true, true
Wscript.Echo "This printer was removed : " & UNCpath
WScript.Quit

' End of Example Script to remove printer
```

Note 1: The first **true**, means force a printer disconnection, whilst the second true means update the user's local profile.

Note 2: Once again, see how the combination of using a variable (UNCpath) and WScript.Echo results in a neater script than just removing the printer with no confirmation.

Example 3: Mapping Multiple Printers

This is a powerful script which creates two printers (or more), without the need of a second script. I will let you into a secret, this script causes more trouble than all the rest put together, so take care with the punctuation and carriage returns.

Instructions

Copy the following script. Now open Notepad, paste in the text, when the script is ready, save the file with a .vbs extension. Example MultiPrinter.vbs

Change the path for each variable to match a server and printer on your network. For example, change "\\Server1\printer1" to a print server on your network.

```
' VBScript to map two printers from two servers
' Guy Thomas February 2004.
' *****
Dim multiPrinter, UNCpath1, UNCpath2
UNCpath1 = "\\Server1\printer1"
UNCpath2 = "\\Server2\printer2"
Set multiPrinter = CreateObject("WScript.Network")
multiPrinter.AddWindowsPrinterConnection UNCpath1
multiPrinter.AddWindowsPrinterConnection UNCpath2
WScript.Echo "Your printer is mapped from : " & UNCpath1 _
& "and from : " & UNCpath2
WScript.Quit
' End of Script
```

Note 1: You can easily add more printers by copying line 9 and pasting a new line 10. Remember to add another UNCpathx statement.

Trap: If you recycle the SAME UNCpath variable, your script will only map one printer. For example this is wrong, it will only map one printer not three.

```
UNCpath1 = "\\Server1\printer1"
UNCpath1 = "\\Server2\printer2"
UNCpath1 = "\\Server3\printer4"
```

Summary

Mapping a network printer is one of the classic jobs for a logon script. When learning, it's handy to have a mirror image - a script to disconnect a printer. Another of my favourite learning techniques is to build up slowly, create a script to map one drive, then add code to map extra drives.

Contents for Ezine 68 - Restart Services

1. **This week's secret**
2. **Example 1 Stop the Service**
3. **Example 2 Start the Service**
4. **Example 3 to Restart the service**
5. **Readers Letters**
6. **Summary**

This week s secret

In the 1970's I had a career, or a 'life' in research. When it came to producing a report or a thesis it was commonplace for we students to write 100s of pages. My mate Rod showed me his thesis and to my amazement, it was only 1 page long. I was even more surprised when his professor said it was the finest piece of work that he had ever seen. What appealed to the professor was the script's clarity and simplicity. Rod had written an original mathematical proof of a theorem that had never been solved before - all on one page.

What has Rod's one page thesis got to do with scripting? Well, last week I asked if anyone knew of utility similar to iisreset which restarts IIS, Hans K wrote in saying, 'Try Net Stop, Net Start'. I have taken Hans's simple but brilliant idea and made it into a script to restart any Windows service.

This week's Mission to restart Windows Services

This week I have yet another script that works at several levels. The practical goal is to stop and then restart Windows services such as, Net Logon, IPSec or System Attendant. Meanwhile, the scripting goals include mastering 'SendKeys' and creating shell objects, (rather than the usual network objects).

Preamble

When ever I experiment with scripts that affect the operating system, my fear is that the script will run out of control and shutdown a vital service. So in the testing phase, I choose services that will have minimal affect if anything goes wrong, for example, Alerter or Messenger, (rather than Net Logon or System Attendant).

Two Tips:

- 1) Try Tasklist from the command prompt. On any of the XP family, Tasklist enumerates the names of services running on that computer.
- 2) Investigate the Services snap-in with a view to identifying other services that you could restart. Only last week, I needed to start Remote Procedure Call (RPC) Locator. Before that I need a script to control Telnet.

Example 1 Stop the Service

Here is your script to stop a Windows service. This example uses Net Stop to halt the Alerter service.

Instructions to Stop a Service

1. Copy and paste the script below into notepad.
2. Save the file with .vbs extension e.g. Services.vbs
3. Double click and then watch a command prompt open and the script stop the Alerter service.
4. The script will then close the cmd prompt and display a message.

```
' Services.vbs
' Example VBScript to Stop Services
' Author Guy Thomas http://computerperformance.co.uk/
' Version 1.3 - March 20th 2005
' -----
Option Explicit
Dim objShell, intShortSleep, intLongSleep
Dim strService
Set objShell = CreateObject("WScript.Shell")

' Values set
strService = " Alerter"
intShortSleep = 1500
intLongSleep = 3500

' Cmd prompt opened
objShell.Run "cmd"
Wscript.Sleep intShortSleep

' Service stopped with 'Net' command
objShell.SendKeys "net stop" & strService
Wscript.Sleep intShortSleep
objShell.SendKeys "{Enter}"
Wscript.Sleep intLongSleep

' Cmd prompt exited
objShell.SendKeys "Exit"
Wscript.Sleep intShortSleep
objShell.SendKeys "{Enter}"

Wscript.Echo strService & " service stopped "
WScript.Quit

' End of Example VBScript
```

Learning Points

Note 1: What we need here is a shell object to hold the command prompt. See Line 9: `Set objShell = CreateObject("WScript.Shell")`

Note 2: The star amongst the scripting command is `SendKeys`. I expect that you have seen how it mimics you typing from the command line.

Note 3: From the operating system's point of view, the vital command is `net stop`.

Challenge: Substitute another service for `Alerter`.

Example 2 Start the Service

The idea here is to build a script which is the mirror image of Example 1, namely to start the service. To achieve this goal just alter these two lines.

```
    ' Service stopped with 'Net' command  
Line 21: objShell.SendKeys "net stop" & strService ' Change to NET START  
  
Line 31: Wscript.Echo strService & " service STARTED "
```

Learning Points

Note 1: Rather than typing out the whole script, I challenge you to change two lines.

Note 2: The key point is to reverse the `net stop` command to `NET START`.

Example 3 Restart the Service

This script combines Example 1 and 2 to create a script which restarts the Windows Alerter service.

```
' RestartServices.vbs
' Example VBScript to Restart Services
' Author Guy Thomas http://computerperformance.co.uk/
' Version 1.3 - March 20th 2005
' -----'
Option Explicit
Dim objShell, intShortSleep, intLongSleep
Dim strService
Set objShell = CreateObject("WScript.Shell")

' Values set
strService = " Alerter"
intShortSleep = 1500
intLongSleep = 5500

' Cmd prompt opened
objShell.Run "cmd"
Wscript.Sleep intShortSleep

' Service stopped with 'Net' command
objShell.SendKeys "net stop" & strService
Wscript.Sleep intShortSleep
objShell.SendKeys "{Enter}"
Wscript.Sleep intLongSleep

' Service started with 'Net' command
objShell.SendKeys "net start" & strService
Wscript.Sleep intShortSleep
objShell.SendKeys "{Enter}"
Wscript.Sleep intLongSleep

' Cmd prompt exited
objShell.SendKeys "Exit"
Wscript.Sleep intShortSleep
objShell.SendKeys "{Enter}"

Wscript.Echo strService & " service stopped "
WScript.Quit

' End of example VBScript
```

Learning Points

Note 1: I have increased the sleep time to 5500 about 5 seconds. You may wish to increase this value to ensure that the service has stopped before the script attempts a net start.

Note 2: If the service is not running, the script will initially produce an error, but then go on to start that service. This effect may or may not be what you desire.

Scripting Challenge: Introduce a MsgBox to prompt for the name of the service you wish to restart.

Windows Challenge: Investigate other services, which you could start and stop with a VBScript. To get ideas, open a command prompt and type: tasklist. Then switch to the Services snap-in for other likely services to restart.

Summary - Restart Services

Here is a handy script to restart your Windows services. The main points are creating a shell object and employing SendKeys to mimic typing from the keyboard.

Contents for Guy's Scripting Ezine 36 - FSO Copy

1. **This week's secret**
2. **Example 1 - Create a text file, then copy it to a new path**
3. **Example 2 - Move File, deleting the old one**
4. **Guy's Challenge - try and solve this 'Out take'**
5. **Summary of FSO**

This week's secret

This week's secrets are the FSO 'dot' commands. What I am referring to are the methods, which you employ to manipulate VBScript objects. For example, once you have created an object called objFSO by using the command:
Set objFSO = CreateObject("Scripting.FileSystemObject"), then you can .Copy, or .Move, and so manipulate that file object. These .dot commands open up a whole new area of scripting and not just for FSO. How do you find out which .dot commands are available for which command?

Example 1 Create a text file, then copy it to a new path.

One task that is ripe for scripting is automating file copying, for example, housekeeping log files or daily accounts records. From a scripting point of view, this weeks example extends our previous lessons with FSO (File Scripting Object) in Ezine 36 and Ezine 12.

Instructions

1. For strFilePath and strDestination I could have used "C:\". Instead, I deliberately left in a path that exists on my machine because I wanted you to think about these string values and then either adjust the script (best) or create the corresponding folders on your machine.
2. Copy and paste the main script into notepad. Alternatively, use a script editor like VBsEdit.
3. Save the file with .vbs extension e.g. CopyFSO.vbs
4. Double click and observe the file that you have just created and copied.
5. If you get errors, check that you have amended strFilePath and strDestination. If you still get Code 800xxxxx message boxes, check out the [error codes](#)

Example 1 - Create a text file, then copy it to a new path

```
' CopyFSO.vbs
' Example VBScript for manipulating files.
' Author Guy Thomas http://computerperformance.co.uk/
' Version 3.4 - 4th July 2004
' -----'
Option Explicit

Dim objFSO, objFileCopy, objGuyFile
Dim strFilePath, strDestination
Dim strFileText, strFileText2, strFileText3

strFilePath = "e:\ezine\scripts\ezine36\newsletter36.txt"
strDestination = "e:\ezine\scripts\ezine36\4thJuly.txt"
strFileText = "Today is American Independence day "
strFileText2 = "Next Ezine 11 July"
strFileText3 = " Bye from Guy"

Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objGuyFile = objFSO.CreateTextFile(strFilePath, True)

objGuyFile.WriteLine(strFileText)
objGuyFile.Write (strFileText2)
objGuyFile.WriteLine (strFileText3)
objGuyFile.Close

WScript.Echo "Written " & strFileText & " to " & strFilePath

Set objFileCopy = objFSO.GetFile(strFilePath)
' Copy the file to its destination
objFileCopy.Copy (strDestination)
WScript.Echo "Copied to " & strDestination
Wscript.Quit

' End of Script
```

Learning Points

Note 1: Set objFSO = CreateObject("Scripting.FileSystemObject") builds our file object. Once we have created objFSO, then we can put it to work and copy or move files.

Note 2: This is script of two halves, the first half creates the file called newsletter36.txt and adds text. The second half, (below the line: WScript.Echo "Written "), copies the file to a new destination.

Note 3: In addition to the .Copy, spot the .GetFile method which is essential to 'grasp' the file before it can be copied or deleted.

Challenge 1.

It would more spectacular and realistic to copy the file to another drive or at least to another folder. So my challenge is to amend your strDestination.

Example 2 - Move File, deleting the old one

Challenge 2.

My idea here is to move the newsletter36.txt file rather than copy it.

Change .Copy to .Move; this is what I mean, alter objFileCopy.Copy (strDestination) to objFileCopy.Move (strDestination)

N.B. You are likely to get an error message, unless you add the .delete command.

```
' MoveFSO.vbs
' Example script to move files.
' Author Guy Thomas http://computerperformance.co.uk/
' Version 4.2 - 4th July 2004
' -----
Option Explicit

Dim objFSO, objFileCopy, objGuyFile, objFileDelete
Dim strFilePath, strDestination
Dim strFileText, strFileText2, strFileText3

strFilePath = "e:\ezine\scripts\ezine36\newsletter36.txt"
strDestination = "e:\ezine\scripts\ezine36\4thJuly.txt"
strFileText = "Today is American Independence day "
strFileText2 = "Next Ezine 11 July"
strFileText3 = " Bye from Guy"

Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objGuyFile = objFSO.CreateTextFile(strFilePath, True)

objGuyFile.WriteLine(strFileText)
objGuyFile.Write (strFileText2)
objGuyFile.WriteLine (strFileText3)
objGuyFile.Close

WScript.Echo "Written " & strFileText & " to " & strFilePath

' Delete file if it already exists in the destination.
Set objFileCopy = objFSO.GetFile(strDestination)
objFileCopy.Delete ()

Set objFileCopy = objFSO.GetFile(strFilePath)
' Move the file destination

objFileCopy.Move (strDestination)
WScript.Echo "Copied to " & strDestination
Wscript.Quit

' End of Script
```

Learning Points

Note 1: If the file already exists then the script would generate an error - unless you add the `.delete` command.

Note 2: The `.delete` command takes a blank `()` argument. The syntax is `objFileCopy.Delete()` not `objFileCopy.Delete(strDestination)`

Note 3: I suspect that there are other ways of moving and overriding the existing file. However adding 'True' to the command for example, `: (strDestination,True)` did not work.

Contents for Guy's Scripting Ezine 28 - inStr (In String)

1. **This week's secret**
2. **Example 1 - inStr function**
3. **Example 2 - Script to check for 'Beth' in a user's name**
4. **Guy's Challenges**
5. **Summary**
6. **Update - New Example of InStr**

This week s secret inStr and Left

There have been many times when I wanted a script to filter objects, for example, suppose you want a list of all users whose name contains Beth - Elizabeth, Bethany, Elsbeth and other derivatives. I have to confess that in the past I used the 'Left' function, whereas I would have been much better off using the 'inStr' function.

Once you realize that inStr is not some old variable that Guy invented, but a proper scripting function, then it is easy to understand what inStr does. inStr literally mean within the string, in a nutshell, inStr is for filtering or selecting part of the string.

Powerful, awkward, tricky and versatile are all words to describe inStr. My contributions to this inStr debate are, 'know the rules' and try lots of examples.

The rules for the function inStr

inStr has the following four components.

InStr([start character,] string searched, string to look for [, Binary or Textual)

- 1) The first parameter is optional and says 'which character should I start at?' If you omit the first parameter, the default is begin with the left most character.
- 2) The parameter says, 'where do I look?' Have you declared a variable? or are you giving me a string value?
- 3) What should I look for? 'Bet' or 'Beth'?
- 4) Lastly another optional component. This says 'is the search at the binary level or is it textual?', in practical terms this means is it case sensitive? In part 3) above, do you mean strictly 'Beth' or do you also want beth'? A value of 1 indicates textual, in plain English this means - not case sensitive.

Example 1: Getting Started - To produce values in inStr

My strategy with inStr is to build up gradually. While this script is of little practical use, it does have interesting learning points and will set the scene for Example 2. Incidentally I wanted a script which would work on any machine.

Instructions

1. Copy and paste the script below into notepad. Alternatively, use a script editor like OnScript.
2. Save the file with .vbs extension e.g. GuyFunction.vbs
3. Double click and observe the message box
4. If you get errors, pay attention to syntax and object names.

```
' GuyFunction.vbs - In string function
' VBScript to return values for an inStr function called GuyFunction
' Author Guy Thomas http://computerperformance.co.uk/
' Version 1.2 - May 9th 2004
' -----'

Option Explicit
Dim GuyFunction, strSearch, strLook

strSearched = "Elizabeth"
strLook = "Beth"

GuyFunction = InStr(1,strSearched, strLook,1)
Wscript.Echo "inStr "& GuyFunction
Wscript.Quit

' End of Script
```

Note 1: GuyFunction = InStr(1,strSearch, strLook,1)

a) The first **1** means start at the first character.

b) The second **1** mean make a textual search. To me, 'textual' means case insensitive. Try changing to zero and see what happens. inStr 0 means that it did not find any occurrence of 'Beth'. Try changing 'Beth', to 'beth' and then running the script again.

Note 2: strSearched is the string to be checked (Elizabeth), while strLook is tells it which letters to look for in strSearched (Beth).

Example 2: Script to check for the string 'Beth' in user's name

In this second example we put inStr to some real work. But firstly, you need to create an OU and three users in your Active Directory. As you create these objects, so you may get ideas for objects that you wish to filter. For example to move computers whose name contains certain letters.

Preparation:

- 1) Go to your Active Directory users and computers. Create an OU called Worcester.
- 2) Create three users called Bethany, Joan and Elizabeth

Instructions for InStr VBScript

1. Follow the preparation above, create the Worcester OU, and the three users in Active Directory.
2. Copy and paste the script below into notepad. Alternatively, use a script editor like VBsEdit.
3. Save the file with .vbs extension e.g. inStr.vbs
4. Double click and observe the message box
5. If you get errors, pay attention to syntax and line numbers.

```
' inStr.vbs - In string function
' Example VBScript to test strings for content
' Author Guy Thomas http://computerperformance.co.uk/
' Version 3.2 - May 9th 2004
' -----'

Option Explicit
Dim objRootDSE, objDomain, objUser, objectclass, user
Dim strDNSDomain, strOU, strName, strPlace, strProperty
Dim intCounter

Set objRootDSE = GetObject("LDAP://RootDSE")
strDNSDomain = objRootDSE.Get("DefaultNamingContext")
strOU = "OU=Worcester,"
strName = "Beth"
strPlace = strOU & strDNSDomain

Set objDomain = GetObject("LDAP://" & strPlace)
For each objUser in objDomain
strProperty=objUser.get("name")
If InStr (1,strProperty, strName, 1) > 0 Then
Wscript.Echo objUser.name
End If
Next

WScript.Quit

' End of Script
```

Expected result: Bethany and Elizabeth.

Learning Points

Note 1: Being a function, `inStr` returns a value. In this example, if the user name contains 'beth' anywhere in the string, then the user's name will be echoed. From the first example you learnt that `inStr` produces values. In this instance, the 'If' test check to see if the value returned is greater than zero.

Note 2: `If InStr (1, strProperty, strName, 1) > 0 Then`

- a) In the above line, the first 1 means start at the first character.
- b) In the above line the second 1 means make a textual search, that is, a non case sensitive comparison.

Challenges for InStr

This week, instead of deliberate mistakes, I have a series of challenges for you. A few tiny changes make all the difference between the script 'working' or producing unexpected results.

1) `If InStr (1, strProperty, strName, 1) > 0 Then`

Try changing this second 1 to zero and see the difference. (Now you only get Bethany and not Elizabeth).

2) Try changing `If InStr (1, strProperty, strName, 1) = 1 Then`

Again you only get Bethany, but the reason is because 'Beth' is at the beginning of the word.

3) Try changing this: `If InStr (1, strProperty, strName, 1) = 6`, and see if you only get Elizabeth. The reason for only getting ElizaBETH, and not Bethany that theBETH starts at the 6th character. The Beth in Bethany starts at the first character.

Summary of InStr

`inStr` is a very useful filter for finding objects in Active Directory. Note that it has four components which control the start point, string to search, value to compare and whether it should be case sensitive.

Contents for Guy's Scripting Ezine 46 - MsgBox

1. **This week's secret**
2. **Example 1 Simple Message Box with one button, OK**
3. **Example 2 Message Box with 3 buttons, Yes, No and Cancel**
4. **Summary**

This week s secret

Early in my scripting career, I thought that MsgBox and WScript.echo were practically one and the same. Wrong. Worse still, I used to believe that when it came to generating message boxes, MsgBox was inferior to WScript.echo - wrong again. Please learn from my mistakes, and open your mind to employing MsgBox, especially in situations where you are dealing with users making choices.

MsgBox

I now realise that in terms of features, MsgBox is superior to WScript.echo. The first point to note is that unlike WScript.echo, MsgBox is a function. The significance is that MsgBox can return a variety of values depending on which dialog button you press. So, MsgBox opens up exciting possibility for scripting, let us study an example.

Example 1 Simple Message Box with one button, OK.

The purpose of this example is to introduce WMI (Windows Management Interface) which will act as a vehicle for our MsgBox experiments.

Scenario, you want a script to provide information about the operating system. We can use WMI command to retrieve information about the computer. For example:

```
Set objWMIService = GetObject("winmgmts: While we could write the results to a file, I would prefer to display the WMI information on screen in a message box.
```

In truth, the MsgBox function is not really needed for our first script. However, please bear with me, because later examples will be more interesting and complex. Remember that this example is better than the usual 'Hello World' introductions.

Instructions

1. Copy and paste the script below into notepad.
2. Save the file with .vbs extension e.g. MessageBox.vbs.
3. Double click and examine the message box.

```
' MessageBox.vbs
' Example VBScript to generate a message box.
' Author Guy Thomas http://computerperformance.co.uk/
' Version 1.3 - September 19th 2004
' -----

Option Explicit
Dim objWMIService, objItem, collItems
Dim strComputer, strMbox
'On Error Resume Next
strMbox = "."

strMbox = MsgBox("Guy says you must select the localhost!")
If strMbox = 1 Then
strMbox = "."
End If

Set objWMIService = GetObject("winmgmts:\\." & strMbox & "\root\cimv2")
Set collItems = objWMIService.ExecQuery("Select * from
Win32_OperatingSystem",,48)

For Each objItem in collItems
WScript.Echo "HostName: " & vbTab & objItem.CSName & vbCr _
& "-----" & vbCr _
& "Operating System: " & vbTab & objItem.Caption & vbCr _
& "Version: " & vbTab & vbTab & objItem.Version & vbCr _
& "System Device: "& vbTab & objItem.SystemDevice & vbCr _
& ""
Next

WScript.Quit
' End of example VBScript
```

Learning Points

Note 1: There is only a single OK Button! Later examples will have the additional buttons: 'No' and 'Cancel'.

Note 2: Numeric 1 is the return value. So the statement:
If strMbox = 1 Then strMbox = "." Sets the WMI script to interrogate the local host. Again this idea of using the return value is to prepare the ground for more complex scripts.

Note 3: The script also employs a loop: For Each Next. Collation items such as objItem.Version are properties of the computer, which are extracted by the WMI part of the script.

Example 2 Message Box with 3 buttons, Yes, No and Cancel

Now it's time to build on the first example and to dissect the MsgBox() function.

```
MsgBox(prompt[, buttons][, title])
```

I will employ the 'Prompt' argument to display text in the Message box. 'Prompt' tells the user what the message box will achieve.

The most interesting MsgBox argument is - 'buttons'. The default value of 0 (zero) means just displays the OK button. If we change the value of buttons from 0 to 3 then we get two extra buttons, 'No' and 'Cancel'.

Example A: Buttons is assumed to be 0, the default value.

```
strMbox = MsgBox("Guy says you must select the LocalHost!")
```

Example B: Buttons is explicitly set to **3** (and title is added).

```
strMbox = MsgBox("Do you want the LocalHost?", 3 ,"Hostname")
```

When scripting with MsgBox, there are two numbers to be aware of, firstly there is the input or display 'Button' number, which controls the actual Message Box. Secondly, there is a separate output or return value number which we do not normally see. However, this return value can be acted upon by subsequent lines of the script.

Naturally, what happens to the output depends on which button you press, this is because unlike WScript.echo, MsgBox() is a function, and functions return values. Clicking the Yes button results in a return value of 6.

To give the script some extra beef, I have introduced an Else statement with an Inputbox which fields all return values that are not equal to 6.

```
' MessageBox3.vbs
' VBScript to generate a message box.
' Author Guy Thomas http://computerperformance.co.uk/
' Version 2.2 - September 19th 2004
' -----

Option Explicit
Dim objWMIService, objItem, collItems
Dim strHost, strMbox
' On Error Resume Next
strMbox = "."

strMbox = MsgBox("Would you like WMI data on localhost?",3,"Hostname")
If strMbox = 6 Then
strHost = "."
Else
strHost = InputBox("Enter your preferred Hostname","Hostname","LocalHost")
End If

Set objWMIService = GetObject("winmgmts:\\\" & strHost & "\root\cimv2")
Set collItems = objWMIService.ExecQuery("Select * from
Win32_OperatingSystem",,48)

For Each objItem in collItems
Wscript.Echo "HostName: " & vbTab & objItem.CSName & vbCr _
& "-----" & vbCr _
& "Operating System: " & vbTab & objItem.Caption & vbCr _
& "Version: " & vbTab & vbTab & objItem.Version & vbCr _
& "Build Number: " & vbTab & objItem.BuildNumber & vbCr _
& "Build Type: " & vbTab & objItem.BuildType & vbCr _
& "Serial Number: "& vbTab & objItem.SerialNumber & vbCr _
& "Product Type: "& vbTab & objItem.ProductType & vbCr _
& "OSLanguage: " & vbTab& objItem.OSLanguage & vbCr _
& "Boot Device: " & vbTab & objItem.BootDevice & vbCr _
& "System Directory: "& vbTab & objItem.SystemDirectory & vbCr _
& "System Drive: " & vbTab & objItem.SystemDrive & vbCr _
& "SPack Major: "& vbTab & objItem.ServicePackMajorVersion & vbCr _
& "SPack Minor: "& vbTab & objItem.ServicePackMinorVersion & vbCr _
& ""
Next

WScript.Quit
' End of Script
```

Learning Points

Note 1: If `strMbox = 6`, then WMI processes the LocalHost and returns its operating system data.

Other return values for `MsgBox()` are 1 = OK, 2 = Cancel, 7 = No. (And of course, 6 = Yes)

Note 2: If the user clicks NO, or Cancel then `strMbox` is not 6, so an Inputbox is displayed.

Challenge: Set the 'Buttons' value to 4. That should just display Yes and No (Removes the Cancel Button).

Here is the line to edit:

```
strMbox = MsgBox("Would you like WMI data on LocalHost?",3,"Hostname")
```

Summary

`MsgBox` is just the ticket for scripts where you need simple user confirmation. Under the covers, the `MsgBox()` function supplies a rich set of scripting input and output values. These return values act as branches for your underlying script.

Contents of Guy's Scripting Ezine 18 - Carriage Returns

1. This weeks secret
2. Carriage returns in VBScripts
3. Example 1 - To find and display all computer names in Active Directory
4. Example 2 - Extract computer names and save into a file
5. Summary

This week s secret

When ever I get an email with complaints it really hurts me. When I recover from the criticism I want to learn from what the person is saying, so I ask myself, 'Is this fair comment? Or is it just some nutter writing in?' Well in the case of those who asked for friendlier printer pages in my ebooks, clearly, our poll results show that you represent what the majority want.

52% Said that printing was invaluable for ebooks.

33% Use printing now and again.

15% Rarely or never used printing.

I am truly grateful for you comments and will endeavour to make all future ebooks printer friendly. A bigger company than mine would have appointed a marketing manager. They would have spent thousands of dollars to tell me what you told me in a minute, I thank all those readers for taking that minute to vote.

Carriage returns in scripts.

Scripts are nothing more than a series of instructions. The VBScript engine needs to know when an instruction ends or when the second half of the instruction continues on then next line.

Early in your VBScript career you may have thought an underscore _ At the end of a line was a dead insect that had crawled on to the page! If you just copy and paste the code, then you may have dismissed these underscores as of no consequence, when in fact they are key to the flow of your script.

When you start writing your own scripts, then you really need to understand how to handle carriage returns. There are two key facts, firstly there has to be a limit of characters per line often 80. Secondly there must be a way of telling the script engine that you have not yet reached the end of a line. Without a word-wrap instruction (`\r`), VBScript would try and interpret every line as a separate statement. The result would be nonsense. If you try and place a normal carriage return at the end of the first line, the scripting engine thinks you now have two lines. The most likely outcome is a WSH error code 800A0401 or 409.

The answer to the word-wrap problem lies in ending the first line with an underscore _ then continue the command on the next line.

Concatenate with ampersand

Concatenate and ampersand are not words you are likely to slip into everyday conversation, however they are vital for the smooth running of your scripts. This weeks learning points are the use of _ (underscore) and & (ampersand) to control carriage returns in the main body of your script. In the case of WScript.Echo commands you can employ additional commands, VBCr (carriage return) and VBTab (to align multiple lines).

Example 1 - Find and display all computer names in Active Directory

The purpose of Example1, is to enumerate and then display all the computer objects in your domain. (Remember to change dc=cp to reflect your domain name).

Copy the VBScript below, then paste into notepad. Crucial point, to get my script to work in your domain, change this line to reflect your Active Directory name:

```
& 'LDAP:// DC=cp, DC=com' _
```

Now you are ready to save the file with .vbs extension, for example, computers.vbs. Double click your computers.vbs file and see all machines in your Active Directory.


```
' Computers.vbs
' Example VBScript to display the Computer Objects in your domain
' Guy Thomas February 2004.
' *****
Set objConnection = CreateObject("ADODB.Connection")
Set objCommand = CreateObject("ADODB.Command")
objConnection.Provider = "ADsDSOObject"
objConnection.Open "Active Directory Provider"
Set objCommand.ActiveConnection = objConnection
objCommand.CommandText = _
"Select Name, operatingSystemVersion from 'LDAP://dc=cp, dc=com' _
& " where objectClass='computer' "
objCommand.Properties("Page Size") = 500
objCommand.Properties("Timeout") = 20
objCommand.Properties("Cache Results") = False
Set objRecordSet = objCommand.Execute
objRecordSet.MoveFirst
Do Until objRecordSet.EOF
Wscript.Echo "Computer: " & objRecordSet.Fields("Name").Value & VbCr _
& "Version: " & objRecordSet.Fields("operatingSystemVersion").Value
objRecordSet.MoveNext
Loop
' End of example VBScript
```

Learning points

1) Making carriage returns and Word-Wrap commands in the script itself.

```
"Select Name, operatingSystemVersion from 'LDAP://dc=cp, dc=com'" _  
& " where objectClass='computer' "
```

The above two lines are responsible for extracting all the computer from your Active Directory. Because they do not comfortably fit on one line, we need a carriage return - THAT VBSCRIPT UNDERSTANDS. Were you to use the normal word processing 'enter', VBScript would interpret the above as two separate statements. So, that is why we need and _ at the end of the first line.

 If you are in doubt, try running the script with the command all on one line.

```
"Select Name, operatingSystemVersion from 'LDAP://dc=cp, dc=com'" & " where  
objectClass='computer' "
```

2) Controlling WScript.Echo messages

```
WScript.Echo "Computer: " & objRecordSet.Fields("Name").Value & VbCr _  
& "Version: " & objRecordSet.Fields("operatingSystemVersion").Value
```

In the above two lines, I have combined two ideas. Firstly, I want to make the script believe this is just one instruction - not two. Secondly, I want the output to be in one box and not two.

As with the previous learning point, _ and & enable us to break the line for presentation purposes. Turning our attention to the output - & VbCr makes a line break WITHIN THE MESSAGE BOX.

Summary:

When you have a long statement, finish the first line with _ and thus join the two lines into one clear statement.

Be clear that VbCr is purely to make the output clearer. Whilst the underscore is essential for the script to execute, VbCr is merely a cosmetic device to make the messages easier to read.

Guy Challenges.

1) Remove **& VbCr** from line beginning WScript.Echo. See how the ComputerName and version run into one another and are all on the same line.

2) Introduce a filter which only display XP machines. What you need to do is add an extra line, find then change:

```
"Select Name, operatingSystemVersion from 'LDAP://dc=cp, dc=com'" _  
& " where objectClass='computer' "
```

To:

```
"Select Name, operatingSystemVersion from 'LDAP://dc=cp, dc=com'" _  
& " where objectClass='computer' " _  
& " and operatingSystemVersion = '5.1 (2600)'"
```

Note the addition of the _ in the middle line, and the & on the third line.

If you have Windows 2000 machines, substitute '5.0 (2195)' for '5.1 (2600)'

3) If you are getting too much output you could try different filtering by altering 'LDAP://dc=cp, dc=com' to 'LDAP://cn=computers, dc=cp, dc=com'

What this does is display all the machines specifically in the Computers folder in Active Directory User's and Computers. If you prefer to specify an OU the command is OU=MyOrgUnit. Note cn=MyOrgUnit will not work.

Example 2 - Extract the names of all computers in your domain, and save to a text file in the C:\

I would like to thank C.K. for sending in the original script.

Note: Remember to find and change this line:

```
'LDAP://DC=cp,DC=com' "_
```

Note: The filename is Computernames.txt

```
' VBScript to extract Computer Objects and save to a text file in C:\
' Guy Thomas February 2004. With credit to CK.
' *****
Const ADS_SCOPE_SUBTREE = 2
Dim objOU
Dim objFileSystem
Dim objComputer
Dim objFileLocation
objFileLocation = "C:\computernames.txt"

Set objFileSystem = CreateObject("Scripting.FileSystemObject")
Set objComputer = objFileSystem.CreateTextFile("objFileLocation", TRUE)
Set objConnection = CreateObject("ADODB.Connection")
Set objCommand = CreateObject("ADODB.Command")
objConnection.Provider = "ADsDSOObject"
objConnection.Open "Active Directory Provider"
Set objCommand.ActiveConnection = objConnection
objCommand.CommandText = "SELECT Name FROM " _
&" 'LDAP://DC=cp,DC=com' "_
& "WHERE objectClass='computer'"
objCommand.Properties("Page Size") = 1000
objCommand.Properties("Timeout") = 30
objCommand.Properties("Searchscope") = ADS_SCOPE_SUBTREE
objCommand.Properties("Cache Results") = False
Set objRecordSet = objCommand.Execute
objRecordSet.MoveFirst
Do Until objRecordSet.EOF
objComputer.WriteLine objRecordset.Fields(0).Value
objRecordSet.MoveNext
Loop
Wscript.Echo "Computer names saved to this path: " & objFileLocation'
' End of Script
```

Learning Point - attention to detail.

Consider the following statement:

```
objCommand.CommandText = "SELECT Name FROM 'LDAP://DC=cp,DC=com' "_
```

Here is how I dissected the line and got it to work:

```
objCommand.CommandText = "SELECT Name FROM " _  
& " 'LDAP://DC=cp,DC=com' "_
```

Here is a tiny difference, lack of " " and it fails

```
objCommand.CommandText = "SELECT Name FROM _  
& 'LDAP://DC=cp,DC=com' "_
```

Const ADS_SCOPE_SUBTREE = 2 this tell the script to drill down two level from the root of Active Directory, in other works to search OUs and subOUs for any computer objects.

Summary of Carriage Returns

Punctuation makes or breaks writing. Once you start placing judicious commas, the reader is able to make sense of your prose. So it is with scripting, VBScript requires special characters to to tell the interpreter that it has reached the end of a line. Take the time to understand _ (underscore) and especially: & (ampersand).

Contents for Guy's Scripting Ezine 50 - pwdLastSet

1. **This week's secret**
2. **This week's theme readers letters**
3. **This week's mission - master pwdLastSet**
4. **Summary**

This week s secret

This time last year, I never thought that we would get to 50 editions of my ezine. Next week I am planning to celebrate with an ebook of all 50 ezines, naturally it will be free. My only worry is making sure that all subscribers get a copy and that is why I set the default ezine to plain text this week. So if you don't like plain text then please edit your preferences.

Last week I could not help noticing that 5 emails in a row came from people whose first language was not English. I felt humble because I cannot speak, let alone write, in French, German, Spanish or Italian. My other feeling was how lucky I am to be raised in a country whose language has been adopted by much of the web.

This week's theme is readers letters.

One of the true joys of editing an ezine is opening the letters that people send in. Your readers' letters really compliment my rough and ready style with my, 'let's get started' mantra. On the one hand, I guess people feel nothing is too simple to write about, while on the other hand people are not shy to point out improvements to my scripts. Here are some examples that I would like to share with you.

Ulrich P has sent in a neat improvement for my Set Password Script. In fact, Ulrich's VBScript is the main feature this week's ezine.

Stéphane has send me the URL of his site which creates custom ActiveX components. I would like to know more of this application of VBscript myself.

Mark P. emailed me with the worlds toughest 800 code error: 0x80041001 "Call failed". This wins the biscuit for the most obscure and least helpful of the code error messages. In fact the whole 080041xxx series are all pretty low on information. I will investigate and post any updates in my Code 800 section of the computerperformance.co.uk web site.

John G. Has finally found a way of mapping printers before users logon. This is his idea:

```
rundll32 printui.dll,PrintUIEntry /b "DisplayPrinterName on http" /x /n "part of the n switch" /if /f %windir%\inf\ntprint.inf /r  
"http://hostname/printers/sharename/.printer" /m "printer name"
```

This week's mission - master pwdLastSet

Suppose you want to force users to change their passwords at next logon. The secret is to employ `pwdLastSet = 0` which triggers Active Directory to display the change password dialog box. One tactic could be that you could email new users with an 'easy' password, but then ensure they secure their user account by choosing a new password which is only known to themselves.

Instructions

1. Important: Which OU will use? My script uses OU=accounts. Either create such an OU, or change the script to reflect YOUR ou=????????.
2. Make sure the OU has 2 or 3 users. If you test the script on a server, grant the users the right to logon locally. (Make them a member of the Server Operators group).
3. Copy and paste the script below into notepad.
4. Save the file with .vbs extension e.g. PwdLastSet.vbs.
5. Double click and examine the message boxes.
6. Attempt to logon as a user in your OU=

```

' PwdLastSet.vbs
' Example VBScript to FORCE a user's password change
' Author Ulrich P amended by Guy Thomas
' Version 2.0 - October 17th 2004
' -----'
Option Explicit
Dim objOU, objUser, objRootDSE
Dim strContainer, strLastUser, strDNSDomain, strPassword
Dim intCounter, intAccValue, intPwdValue

Set objRootDSE = GetObject("LDAP://RootDSE")
strDNSDomain = objRootDSE.Get("DefaultNamingContext")
intAccValue = 544
intPwdValue = 0

' -----'
' Important change OU= to reflect your domain
' -----'

strContainer = "OU=Accounts, "
strPassword = "P@ssw0rd"

' -----'
' Makes the user change P@ssw0rd at first logon
' -----'

strContainer = strContainer & strDNSDomain

set objOU =GetObject("LDAP://" & strContainer )
intCounter = 0
For each objUser in objOU
If objUser.class="user" then
objUser.SetPassword strPassword
objUser.SetInfo

objUser.Put "pwdLastSet", intPwdValue
objUser.SetInfo

objUser.Put "userAccountControl", intAccValue
objUser.SetInfo
intCounter = intCounter +1
strLastUser = objUser.Get ("name")
End if
next

WScript.Echo strPassword & " is Password. UserAccountValue = " _
& intAccValue
WScript.Quit

' End of Script

```

Learning Points

Note 1: The change password part of the script was featured in Ezine 23, I have now added an extra section about UserAccountControl Values.

Note 2: In order to work properly, UserAccountControl = 544 needs the pwdLastSet = 0. Do feel free to experiment with other settings, and combinations. If you find anything interesting do let me know.

Note 3: If you are troubleshooting, particularly check strContainer = "OU=Accounts, " Also if you are testing on a server, ensure that the user can logon locally.

Note 4: Try the script without these two line. (I use an ' [apostrophe] to rem out the lines.

```
objUser.Put "pwdLastSet", intPwdValue  
objUser.SetInfo
```

Note 5: When you set pwdLastSet to zero it forces users to change their passwords at next logon.

Note 6: Admire the construction: For Each... next.

Note 7: The statement: If objUser.class="user" then.. is designed to filter out users from other objects such as computers. Did you spot the End if?

Note 8: Experiment with other WScript.Echo commands.

Summary pwdLastSet

Here is a useful addition to your password changing toolkit - pwdLastSet. What this extra command does is force users to change their password at next logon. Naturally to get the user to change their password, you need to set a value of zero.

Contents for Guy's Scripting Ezine 45 - Shutdown

1. This week's secret
2. Example 1 Script to shutdown a computer
3. Summary

This week s secret

As you may know, I am not a great one for disclaimers. Moreover, the whole nature of scripting is such that occasionally, mistakes just happen. In fact, the road to a successful script is littered with cast-offs. Therefore, it may come as a surprise when I say: 'This week take great care, this script could annoy people, or worse, you could shoot yourself in the foot and shutdown your own machine unexpectedly.' I will let you into a secret, I often forget the /s switch, which is vital for a shutdown and as a result my first effort does not work.

Example 1 Script to Shutdown a Computer

Scenario

Suppose that you need to restart a machine in a distant office. By the time you walk down the corridor and reach the server room, you would like the domain controller to have rebooted.

What we are going to do is employ a VBScript to open up a DOS box, then instruct sendkeys to type in a command which will restart a computer. But first, in order to get the correct machine name, the script generates an input box where you enter the name of the machine you wish to shut down.

The next point that I want to make is this, in the example, I have deliberately left in a VBScript command that will abort the restart. When you are ready to test YOUR script, and for real, remove the lines underneath :

' Abort Shutdown while testing Rem out next x lines in real script

Strongly recommended.

Before you rush off and try the script, take the time to familiarise yourself with the rich variety of 'shutdown' switches. Go to the CMD prompt on a Windows Server 2003 or XP computer and then type shutdown /? For instance, be aware that this command will either shutdown or restart. Furthermore shutdown works on either a local or a remote machine. Naturally, you need to be an administrator to execute shutdown successfully.

Here is your 'Get out of jail card'. WHEN TESTING, if the script does NOT abort the shutdown, you have 60 seconds manually cancel by typing:
Shutdown /a /m \\ computername.

My strongest advice is master the above command FIRST, before you proceed to test shutdown in anger.

Instructions

1. Copy and paste the script below into notepad. Alternatively, use a script editor like VBsEdit.
2. Save the file with .vbs extension e.g. Shutdown.vbs.
3. Double click and then enter in the input box, the computername that you wish to shut down.
4. My VBScript as written, will abort the shutdown, remove the relevant lines when YOU are ready to go live.

```
' Shutdown.vbs
' Example VBScript to Shutdown computers
' Author Guy Thomas http://computerperformance.co.uk/
' Version 3.3 - September 12th 2004
' -----

Option Explicit
Dim objShell, strComputer, strInput, intShort, intLong
Dim strShutdown, strShutdownSwitch, strAbort
' Set the delay for commands in the Dos Box
intShort = 100
intLong = 10000 ' Reduce if you wish

' Input Box to get name of machine to shutdown
Do
strComputer = (InputBox(" ComputerName to shutdown", "Computer Name"))
  If strComputer <> "" Then
    strInput = True
  End if
Loop until strInput = True

strShutdown = "shutdown /s /t 60 /m \" & strComputer
' chr(34) escapes speech marks - so they display in Dos Box
strShutdownSwitch = " /c " & Chr(34) & "Guy Shutdown " & Chr(34) & " /f"
strAbort = "shutdown /a /m \"
Set objShell = CreateObject("WScript.Shell")
objShell.Run "cmd"
Wscript.Sleep intShort
objShell.SendKeys strShutdown & strShutdownSwitch
Wscript.Sleep intLong
objShell.SendKeys "{ENTER}"

' Abort Shutdown while testing Rem out next 8 lines in real script
objShell.Run "cmd"
Wscript.Sleep intShort
objShell.SendKeys strAbort & strComputer
Wscript.Sleep intShort
objShell.SendKeys "{ENTER}"
Wscript.Sleep intShort
' This section closes the Two Dos boxes
Call CloseBox()

Call CloseBox()
```

```
' Sub Routine CloseBox()
Sub CloseBox()
  Wscript.Sleep intLong
  objShell.SendKeys "exit"
  objShell.SendKeys "{ENTER}"
  Wscript.Sleep intShort
End Sub

WScript.Echo "Beware: you tried to shutdown " & strComputer
Wscript.Quit

' End of example Shutdown VBScript
```

Learning Points

Note 1: InputBox is the VBScript method which prompts you for the computer name. This is the name of the machine to shutdown. In effect InputBox saves having to hardcode strComputer = "BigServer". My basic error checking invokes the 'if' command which in turn, checks for blank names. Did you like the 'DO ... Loop Until' command?

Note 2: Sub (routine) is this week's new command. I have experimented with recycling code. In particular we need two DOS boxes, therefore we need two commands to quit the DOS box. Look out for a Sub Function called CloseBox(). You could have used any name, for example, DosExit() would work just as well. How could you take advantage of SUB CloseBox()? The answer is that you 'Call' the sub routine, for example: Call CloseBox(). Note: 'Call Sub CloseBox()', would be wrong.

Note 3: & Chr(34) is used to generate a speech mark sometimes called a double quotes. I needed this trick to make quotes within quotes work successfully.

Note 4: You did remember to remove or 'Rem out the abort lines on the real script?

Note 5: Shutdown /t 60 adds a one minute delay before shutdown. The /c switch is purely an indulgence which adds a comment.

Challenges and Ideas.

1) Add a default value to the inputbox. For example, "LocalHost" or,"BigServer".

```
strComputer = (InputBox("Enter ComputerName to shutdown", "Computer  
Name", "LocalHost"))
```

2) My script employs the usual range of string and integer variables. Perhaps you would like to reduce the value of intLong from 10 seconds to 3 seconds? If so, reduce intLong from 10000 to = 3000.

3) Experiment with different values for the /c "Comment" and /t 60 switches.

4) If you are feeling bolder, why not investigate other shutdown switches, for example /r to restart instead of /s which merely shuts down the machine.

Summary

If you want to save that long walk to the server room then consider the built-in shutdown command. Beware of rebooting the wrong machine! Take the time to investigate the switches of the shutdown command. In particular become expert at the abort switch: shutdown /a /m \\ machinename. The sample script has a 'safety catch' which aborts the shutdown, find the lines and remove in your final production VBScript.

Index of Top 10 Ezines

Guy s Top 10 Ezines	1
Contents (In order of popularity)	1
Contents Ezine 26 - Run, SendKeys and Sleep	2
This week s secret - Everything has a purpose	2
Example 1 - Creating a VBscript to open Notepad	2
Learning Points	3
Challenge:	4
Summary:	4
Contents for Ezine 23 - Enabling User Accounts	5
Please note:	5
This week s secret - CSVDE does not set passwords	5
A Script to Enable accounts that have been 'Disabled' X	5
Learning Points	7
Troubleshooting	7
UserAccountControl Values	7
Force users to change password at next logon	8
Learning Points	8
Contents of Ezine No 16 - Printer Scripts	9
This week's secret	9
Example 1: Mapping your Printer	9
Error Codes generated by this printer logon scripts	10
Example 2 Delete a Printer with - RemovePrinterConnection	10
Example 3: Mapping Multiple Printers	11
Summary	11
Contents for Ezine 68 - Restart Services	12
This week s secret	12
This week's Mission to restart Windows Services	12
Example 1 Stop the Service	13
Learning Points	14
Example 2 Start the Service	14
Learning Points	14
Example 3 Restart the Service	15
Learning Points	16
Summary - Restart Services	16
Contents for Guy's Scripting Ezine 36 - FSO Copy	17
This week's secret	17
Example 1 Create a text file, then copy it to a new path.	17
Example 1 - Create a text file, then copy it to a new path	18
Learning Points	18
Example 2 - Move File, deleting the old one	19
Learning Points	20
Contents for Guy's Scripting Ezine 28 - inStr (In String)	21
This week s secret inStr and Left	21
Example 1: Getting Started - To produce values in inStr	22
Example 2: Script to check for the string 'Beth' in user's name ...	23

Expected result: Bethany and Elizabeth.....	23
Learning Points.....	24
Challenges for InStr	24
Summary of InStr	24
Contents for Guy's Scripting Ezine 46 - MsgBox.....	25
This week s secret.....	25
MsgBox	25
Example 1 Simple Message Box with one button, OK.....	25
Learning Points.....	26
Example 2 Message Box with 3 buttons, Yes, No and Cancel.....	27
Learning Points.....	29
Summary	29
Contents of Guy's Scripting Ezine 18 - Carriage Returns.....	30
This week s secret.....	30
Carriage returns in scripts.....	30
Example 1 - Find and display all computer names in Active Directory	31
Learning points.....	32
Summary:	32
Guy Challenges.....	33
Example 2 - Extract the names of all computers in your domain, and save to a text file in the C:\.....	34
Learning Point - attention to detail.....	35
Summary of Carriage Returns.....	35
Contents for Guy's Scripting Ezine 50 - pwdLastSet.....	36
This week s secret.....	36
This week's theme is readers letters.....	36
This week's mission - master pwdLastSet.....	37
Learning Points.....	39
Summary pwdLastSet.....	39
Contents for Guy's Scripting Ezine 45 - Shutdown.....	40
This week s secret.....	40
Example 1 Script to Shutdown a Computer.....	40
Learning Points.....	42
Challenges and Ideas.....	43
Summary	43