88993

Miscellaneous code I use at home and work

VBScript ****

Author:

[**Michael D. Newby**](http://www.recursivecreativity.com/)

[www.recursivecreativity.com](http://www.recursivecreativity.com)

[www.excelusersgroup.org](http://www.excelusersgroup.org)

(not active)

Last Updated:

**August 2, 2021**

Table of Contents

[Kill Applications, Processes 4](#_Toc78832490)

[File, Folder Counts 5](#_Toc78832491)

[Map Network Drive 7](#_Toc78832492)

[Unmap Network Drive 8](#_Toc78832493)

Each heading below is link back to this page, the above Table of Contents (TOC).

**PLEASE NOTE:**

All formatting of code is off due to be copied from Notepad, which is used to create and edit VBScript and JavaScript files.

All .vbs files are included as part of .zip files since .vbs file are usually blocked due to possible malicious content.

# [Kill Applications](#TOC), Processes

The VBScript code below will kill all instances of Excel that are running.

**PLEASE NOTE:**

Before running any such script, if possible, please make sure all open files have been saved.



The above embedded file(s) can be copied and used for any program or process.

Highlighted in code below is the variable which holds value of application being killed or terminated by the script. You only need to change that for script to be used for other applications or processes.

Option Explicit

'On Error Resume Next

Dim strProcess: strProcess = "**Excel.exe**"

Dim objWMIService: Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate}" & "!\\.\root\cimv2")

Dim colProcess: Set colProcess = objWMIService.ExecQuery ("Select \* From Win32\_Process")

Dim objWshShell: Set objWshShell = CreateObject( "WScript.Shell" )

Dim objProcess

For Each objProcess in colProcess

If LCase(objProcess.Name) = LCase(strProcess) Then

'The next two steps may seem like overkill (pun intended) but they ensure success

objWshShell.Run "TASKKILL /F /T /IM " & objProcess.Name, 0, False

objProcess.Terminate()

'MsgBox "- ACTION: " & objProcess.Name & " terminated"

End If

Next

# [File, Folder Counts](#TOC)

The following script will provide file and folder count for given path, currently highlighted to show what needs to be changed.



Option Explicit

Dim strDirectory, counter, strExtension, msgtext

Dim numFiles, numFolders

Dim objFSO, objFolder, objFile

'file extension to look for

'strExtension = "xlsx"

'directory to look in

strDirectory = "**{Your Drive Letter Goes Here}**:\**{Your Path Goes Here}**"

Set objFSO = CreateObject("Scripting.FileSystemObject")

'check to see if above path\folder exists

If objFSO.FolderExists(strDirectory) Then

Set objFolder = objFSO.GetFolder(strDirectory)

numFiles = objFolder.Files.Count

numFolders = objFolder.SubFolders.Count

msgbox("The counts for given path:" & vbCrLf & vbCrLf & strDirectory & vbCrLf & vbCrLf & "Folders: " & numFolders & vbCrLf & "Files: " & numFiles)

Else

'if NO, provide notification and end script

msgbox("The path or folder does not exist:" & vbCrLf & vbCrLf & strDirectory)

WScript.Quit

End If

I use this script to quickly see if new budget, forecast, Power Query, etc. templates have been posted on SharePoint or network drives. This only works for SharePoint if the SharePoint folder has been mapped as network drive (assigned a drive letter).

The below embedded file is a more complicated version that gets counts for two different paths for comparative purposes.



The above file is being provided as is but must be changed to valid paths before it will work.

# [Map Network Drive](#TOC)

The following script can be used to map network drive. Example assigns Drive letter E: Highlighted items are variables that need to be changed.



Option Explicit

Dim strDriveLetter: strDriveLetter = "**E:**"

Dim strNetworkDrive: strNetworkDrive = "**\\Server\Path**"

Dim objNetwork: Set objNetwork = WScript.CreateObject("WScript.Network")

Dim CheckDrive: Set CheckDrive = objNetwork.EnumNetworkDrives()

Dim bolDriveExists: bolDriveExists = False

Dim strMessage

Dim result

Dim i

For i = 0 to CheckDrive.Count - 1

If CheckDrive.Item(i) = strDriveLetter Then

bolDriveExists = True

End If

Next

If bolDriveExists = False Then

objNetwork.MapNetworkDrive strDriveLetter, strNetworkDrive, True

msgbox "You have mapped Drive " & strDriveLetter & "."

Else

result = MsgBox ("Drive " & strDriveLetter & " is already mapped." & vbcrlf & vbcrlf & "Do you want to unmap Drive " & strDriveLetter & "?", vbYesNo, "Unmap Drive?")

Select Case result

Case vbYes

objNetwork.RemoveNetworkDrive strDriveLetter, True, True

msgbox "You have unmapped Drive " & strDriveLetter & vbCrLf & vbCrLf & "The mapped drive will disappear when this computer is rebooted."

Case vbNo

msgbox "You have chosen not to unmap Drive " & strDriveLetter

End Select

End If

I use multiple versions of above script to assist new employees map network drives.

# [Unmap Network Drive](#TOC)

The following script can be used to unmap network drive. Example unmaps Drive letter E: Highlighted item is variable that needs to be changed.



Option Explicit

Dim strDriveLetter: strDriveLetter = "**E:**"

Dim objNetwork: Set objNetwork = WScript.CreateObject("WScript.Network")

Dim strMessage

objNetwork.RemoveNetworkDrive strDriveLetter, True, True

If Err.Number <> 0 then

MsgBox "Mapped drive, " & strDriveLetter & ", does not exist."

Wscript.Quit

End If

On Error GoTo 0 ' this resets error handling to normal

MsgBox "You have unmapped Drive " & strDriveLetter