

Delete Profiles Script v1.9

Contributed by Joe Shonk
Friday, 29 August 2008

Hello, I am happy to announce that v1.9 of the Delete Profiles script is ready for release. Two new features have been added since v1.8.

New Features:

Inclusion\Exclusion List with wildcard support
Delete Profiles based off of Profile age

The inclusion list specifies which profile or profiles you want to delete. A specific profile can be specified or a wildcard can be used. Multiple profiles can be specified by placing a , (comma) or | (pipe) between each entry.

For example:

```
cscript.exe DeleteProfiles.vbs /I bob,*_admin,xx*,*q*
```

This example will delete the profile for Bob, any profile that ends with _admin, any profile starts with xx and to top it off we are going to delete any profile that has the letter q in it. All other profiles will remain.

We can change this to an exclusion list by replacing the /I switch with /E.

For example:

```
cscript.exe DeleteProfiles.vbs /E bob,*_admin,xx*,*q*
```

Now we have told the script to do the exact opposite. It will delete all profiles except for Bob, those ending in _admin, those starting with xx and any with the letter q in the name.

Note that the Inclusion and Exclusion switches are mutually exclusive. You cannot specify both at the same time. Also, profile names are not case sensitive. If you specify /E BOB the script will treat the item the same as if you specified /E bob.

The days retention switch is fairly simple. If you only want to delete profiles older than a specified number of days then use the /D xx where xx will be in number of days. The days retention switch can also be use in combination with the Inclusion and Exclusion list.

For example:

```
cscript.exe DeleteProfiles.vbs /I bob,joe /D 10
```

Only Bob's and Joe's profile will be delete provided they are older than 10 days. If we change this to an Exclusion list then all profiles older that 10 days will be deleted with the exception of Bob and Joe.

Comments and suggestion can be sent to joe@theshonkproject.com

Syntax for DeleteProfiles.vbs v1.9:

```
cscript.exe DeleteProfiles.vbs [/H] [/E | /I <PROFILENAME>] [/C] [/R] [/D <DAYS>] [/L <FILENAME>] [/V]
```

Command Line Options:

/C : Write Log to the Console
/D <Days> : Delete Profiles Older than x Days
/E <Profile> : Exclude Profiles from Deletion
: Wildcard * Supported. Use ',' or '|' as a Delimiter for
: Multiple Entries. No Spaces Between Entries.
/I <Profile> : Only Delete Included Profiles (Wildcard * Supported)
: Wildcard * Supported. Use ',' or '|' as a Delimiter for
: Multiple Entries. No Spaces Between Entries.
/L <FileName> : Create Log File
/H : Help (This Screen)
/R : Run Script in Read Only Mode (No System Changes)
/V : Verbose Logging

A scheduled task can be used to launch the script as the script will skip any profile hive that is loaded in memory. While this works, it is recommended to launch this script as part of the system startup script (this can be set as a local policy or a GPO). This allows the script to run on a freshly restarted system with no user profiles loaded in memory. It is also recommended that the systems that you run DeleteProfile.vbs against that you also install Microsoft's UPHClean. This will help those situations where the user hive is still loaded but the user is not logged onto the machine.

```

DeleteProfiles.vbs v1.9!*****
!*
!* Delete Profiles script written by Joe Shonk (joe@theshonkproject.com)
!* Version 1.9
!*
!* Syntax: cscript.exe DeleteProfiles.vbs [/H] [/E | /I <PROFILENAME>] [/C] [/R]
!*         [/D <DAYS>] [/L <FILENAME>] [/V]
!*
!* This script is provided as-is, no warranty is provided or implied.
!* The author is NOT responsible for any damages or data loss that may occur
!* through the use of this script. Always test, test, test before
!* rolling anything into a production environment.
!*
!* This script is free to use for both personal and business use, however,
!* it may not be sold or included as part of a package that is for sale.
!*
!* A Service Provider may include this script as part of their service
!* offering/best practices provided they only charge for their time
!* to implement and support.
!*
!* For distribution and updates go to: http://www.theshonkproject.com
!*
*****
On Error Resume Next

Const DeleteReadOnly = TRUE
Const HKEY_LOCAL_MACHINE = &H80000002
Const SIDExclusionList = "|S-1-5-18|S-1-5-19|S-1-5-20|"

*****
!* To add your own profiles to the exclusion list simply add the
!* account to the end of the strProfileExclusionList. Note: Each account
!* is delimited by a | (pipe) and is all lowercase
!*

Dim strProfileExclusionList
Dim strComputer, strLogFileName, strDocAndSettingsLocation
Dim strKeyPath, arrValueNames, arrValueTypes, arrSubKeys
Dim i, strHiveExclusionList, strHiveOpenSkipped, strHiveValue
Dim strSubKey, strGuid, strUserName, strProfileImagePath
Dim dwProfileExclusion, dwSIDExclusion, dwHiveOpenExclusion
Dim flgLogFile, flgWriteConsole, flgVerboseLog, flgAllowExecute, flgHelp
Dim dwArgCount, strNextArg, strCurrentArg, flgCustomExclusions, flgCustomExMatch
Dim strCustomExclusions, dateCurrentTime, flgCustomInclusions, strCustomInclusions
Dim flgCustomInMatch, flgExDateMatch, flgDateExclusion, dwDateExclusion

strComputer = "."
strProfileExclusionList = "|administrator|all users|default
user|localservice|networkservice|public|ctx_smauser|ctx_cpuuser|ctx_cpvcuser|ctx_streamingsvc|ctx_configmgr|"

Set objReg = GetObject("winmgmts:{impersonationLevel=impersonate}!\\" & strComputer & "\root\default:StdRegProv")
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objArgs = WScript.Arguments

strLogFileName = ""
dateCurrentTime = now()
dwArgCount = 0
flgHelp = False

```

```

flgLogFile = False
flgCustomExclusions = False
flgWriteConsole = False
flgVerboseLog = False
flgAllowExecute = True

dwArgCount = objArgs.Count
For i = 0 to dwArgCount - 1
    strCurrentArg = lcase(objArgs(i))
    Select Case strCurrentArg
        Case "-v", "-verbose", "/v", "/verbose"
            flgVerboseLog = True
        Case "-c", "-console", "/c", "/console"
            flgWriteConsole = True
        Case "-r", "-readonly", "-read", "/r", "/readonly", "/read"
            flgAllowExecute = False
            flgWriteConsole = True
        Case "-l", "-log", "/l", "/log"
            If i < (dwArgCount - 1) then
                strNextArg = lcase(objArgs(i + 1))
                If (left(strNextArg, 1) <> "/") and (left(strNextArg, 1) <> "-") then
                    flgLogFile = True
                    strLogFileName = strNextArg
                    i = i + 1
                Else
                    wscript.echo "Warning: Log Switch Used but No Log Filename Specified."
                End if
            Else
                wscript.echo "Warning: Log Switch Used but No Log Filename Specified."
            End if
        Case "-e", "-exclude", "/e", "/exclude"
            If i < (dwArgCount - 1) then
                strNextArg = lcase(objArgs(i + 1))
                If (left(strNextArg, 1) <> "/") and (left(strNextArg, 1) <> "-") then
                    flgCustomExclusions = True
                    strCustomExclusions = replace(strNextArg, ",", "|", 1)
                    i = i + 1
                Else
                    wscript.echo "Error: Exclude Switch Used but no Argument Specified."
                    flgHelp = True
                End if
            Else
                wscript.echo "Error: Exclude Switch Used but no Argument Specified."
                flgHelp = True
            End if
        Case "-i", "-include", "/i", "/include"
            If i < (dwArgCount - 1) then
                strNextArg = lcase(objArgs(i + 1))
                If (left(strNextArg, 1) <> "/") and (left(strNextArg, 1) <> "-") then
                    flgCustomInclusions = True
                    strCustomInclusions = replace(strNextArg, ",", "|", 1)
                    i = i + 1
                Else
                    wscript.echo "Error: Include Switch Used but no Argument Specified."
                    flgHelp = True
                End if
            Else
                wscript.echo "Error: Include Switch Used but no Argument Specified."
                flgHelp = True
            End if
        Case "-d", "-days", "-day", "/d", "/days", "/day"
            If i < (dwArgCount - 1) then
                strNextArg = lcase(objArgs(i + 1))
                If (left(strNextArg, 1) <> "/") and (left(strNextArg, 1) <> "-") then
                    dwDateExclusion = cdbl(strNextArg)

```

```

        i = i + 1
        If (dwDateExclusion > 0) and (dwDateExclusion < 999999) then
            flgDateExclusion = True
        Else
            flgHelp = True
            wscript.echo "Error: Invalid Number of Days Specified."
        End if
    Else
        wscript.echo "Error: Day Switch Used but no Argument Specified."
        flgHelp = True
    End if
Else
    wscript.echo "Error: Day Switch Used but no Argument Specified."
    flgHelp = True
End if
Case "-h", "-help", "/h", "/help", "-?", "/?"
    flgHelp = True
Case Else
    wscript.echo "Unrecognized option: " & objArgs(i)
    flgHelp = True
End Select
next
If flgCustomExclusions and flgCustomInclusions then
    wscript.echo "Error: Cannot Specify a Custom Exclusion List with a Custom Inclusion List."
    flgHelp = True
End if

If flgHelp then
    wscript.echo "Help"
    wscript.echo ""
    wscript.echo "DeleteProfiles.vbs - v1.9"
    wscript.echo "-----"
    wscript.echo ""
    wscript.echo "cscript.exe DeleteProfiles.vbs [/H] [/E | /I <PROFILENAME>] [/C] [/R]"
    wscript.echo "  [ /D <DAYS>] [/L <FILENAME>] [/V]"
    wscript.echo ""
    wscript.echo "Command Line Options:"
    wscript.echo "  /C          : Write Log to the Console"
    wscript.echo "  /D <Days>   : Delete Profiles Older than x Days"
    wscript.echo "  /E <Profile> : Exclude Profiles from Deletion"
    wscript.echo "  /I <Profile> : Wildcard * Supported. Use ',' or '|' as a Delimiter for"
    wscript.echo "              : Multiple Entries. No Spaces Between Entries."
    wscript.echo "  /L <Profile> : Only Delete Included Profiles (Wildcard * Supported)"
    wscript.echo "  /L <Profile> : Wildcard * Supported. Use ',' or '|' as a Delimiter for"
    wscript.echo "              : Multiple Entries. No Spaces Between Entries."
    wscript.echo "  /L <FileName> : Create Log File"
    wscript.echo "  /H          : Help (This Screen)"
    wscript.echo "  /R          : Run Script in Read Only Mode (No System Changes)"
    wscript.echo "  /V          : Verbose Logging"
    wscript.echo ""
    wscript.quit
End if

If flgLogFile then Set objLogFile = objFSO.CreateTextFile(strLogFileName)

WriteHeader

*****
* Enumerate a list of loaded Registry Hives. Delimited by the | character
strHiveExclusionList = ""
strHiveOpenSkipped = ""
strKeyPath = "SYSTEM\CurrentControlSet\Control\hivelist"
objReg.EnumValues HKEY_LOCAL_MACHINE, strKeyPath, arrValueNames, arrValueTypes
For i=0 To UBound(arrValueNames)
    strHiveValue = trim(arrValueNames(i))

```

```
strHiveExclusionList = strHiveExclusionList & Right(strHiveValue, len(strHiveValue) - instrrev(strHiveValue, "\")) & "|"
Next
```

```
*****
```

```
* Enumerate a list of known profiles from the registry
strKeyPath = "SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList"
objReg.EnumKey HKEY_LOCAL_MACHINE, strKeyPath, arrSubKeys
```

```
*****
```

```
* Parse through the Profile list and Delete the Registry entries and Files associated to the Profile
* Provided the profile is not listed in an Exclusion list
WriteLog "Checking Profile List"
WriteLog "-----"
If NOT flgAllowExecute then WriteLog "READ ONLY MODE. No changes made."
If flgDateExclusion then WriteLog "Purge Profiles Older than " & dwDateExclusion & " Days."
WriteLog ""
```

```
For Each subkey In arrSubKeys
```

```
strSubKey = ""
strGuid = ""
strUserName = ""
strProfileImagePath = ""
strSubKey = trim(subkey)
If (instr(SIDExclusionList, "|" & strSubKey & "|") = 0) and (strSubKey <> "") then
strKeyPath = "SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList\" & strSubKey
objReg.GetStringValue HKEY_LOCAL_MACHINE, strKeyPath, "Guid", strGuid
objReg.GetStringValue HKEY_LOCAL_MACHINE, strKeyPath, "ProfileImagePath", strProfileImagePath
```

```
strUserName = Right(strProfileImagePath, len(strProfileImagePath) - instrrev(strProfileImagePath, "\"))
WriteLog "Profile"
If flgVerboseLog then WriteLog " SID      : " & strSubKey
If flgVerboseLog then WriteLog " GUID    : " & strGuid
WriteLog " Profile Path: " & strProfileImagePath
WriteLog " Username  : " & strUserName
```

```
dwProfileExclusion = instr(strProfileExclusionList, "|" & trim(lcase(strUserName)) & "|")
dwSIDExclusion = instr(strHiveExclusionList, "|" & strSubKey & "|")
```

```
If flgCustomExclusions then
flgCustomExMatch = TestCase(lcase(strUserName), lcase(strCustomExclusions))
```

```
    If flgCustomExMatch then
strProfileExclusionList = strProfileExclusionList & trim(lcase(strUserName)) & "|"
dwProfileExclusion = 1
    End if
End if
```

```
If flgCustomInclusions then
flgCustomInMatch = TestCase(lcase(strUserName), lcase(strCustomInclusions))
```

```
    If flgCustomInMatch = 0 then
strProfileExclusionList = strProfileExclusionList & trim(lcase(strUserName)) & "|"
dwProfileExclusion = 1
    End if
End if
```

```
If flgDateExclusion then
flgExDateMatch = 0
If objFSO.FileExists(strProfileImagePath & "\NTUSER.DAT") then
Set objFile = objFSO.GetFile(strProfileImagePath & "\NTUSER.DAT")
flgExDateMatch = TestDateExclusion(objFile.DateLastModified, DateCurrentTime, dwDateExclusion)
Else
WriteLog " NTUSER.DAT Does not Exist"
If objFSO.FileExists(strProfileImagePath & "\NTUSER.MAN") then
```

```

        WriteLog " NTUSER.MAN Found"
        Set objFile = objFSO.GetFile(strProfileImagePath & "\NTUSER.MAN")
        flgExDateMatch = TestDateExclusion(objFile.DateLastModified, DateCurrentTime, dwDateExclusion)
    Else
        WriteLog " NTUSER.MAN Does not Exist"
    End if
End if
If flgExDateMatch then
    strProfileExclusionList = strProfileExclusionList & trim(Icase(strUserName)) & "|"
    dwProfileExclusion = 1
End if
End if
If (dwProfileExclusion = 0) and (dwSIDEExclusion = 0) then
    WriteLog " Profile OK to Delete"
    If flgDateExclusion then WriteLog " Profile Matches Age Requirement"
    If flgCustomInclusions and flgCustomInMatch then WriteLog " Profile Matches Inclusion List"

    strKeyPath = "SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList\" & strSubKey
    DeleteKey HKEY_LOCAL_MACHINE, strKeyPath

    strKeyPath = "SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\" & strSubKey
    DeleteKey HKEY_LOCAL_MACHINE, strKeyPath

    strKeyPath = "SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\State\" & strSubKey
    DeleteKey HKEY_LOCAL_MACHINE, strKeyPath

    strKeyPath = "SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\" & strSubKey
    DeleteKey HKEY_LOCAL_MACHINE, strKeyPath

    If strGuid <> "" then
        strKeyPath = "SOFTWARE\Microsoft\Windows NT\CurrentVersion\PolicyGuid\" & strGuid
        DeleteKey HKEY_LOCAL_MACHINE, strKeyPath

        strKeyPath = "SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileGuid\" & strGuid
        DeleteKey HKEY_LOCAL_MACHINE, strKeyPath
    Else
        WriteLog " Guid is Blank, Deleting Registry Keys based of Guid has been skipped."
    End if

    If objFSO.FolderExists(strProfileImagePath) then
        WriteLog " Folder Exists - Deleting"
        If flgAllowExecute then objFSO.DeleteFolder(strProfileImagePath), DeleteReadOnly
    Else
        WriteLog " Folder Does not Exist"
    End if
Else
    If dwProfileExclusion then
        WriteLog " Profile not Deleted --- Username in Profile Exclusion List"
        If flgCustomExclusions and flgCustomExMatch then _
            WriteLog " Profile not Deleted --- Username Matched Custom Exclusion List"
        If flgCustomInclusions and (flgCustomInMatch = 0) then _
            WriteLog " Profile not Deleted --- Username Did not Match Custom Inclusion List"
        If flgDateExclusion and flgExDateMatch then _
            WriteLog " Profile not Deleted --- Profile is not Older than " & dwDateExclusion & " Days"
    End if
    If dwSIDEExclusion then
        WriteLog " Profile not Deleted --- User Hive is currently loaded"
        strHiveOpenSkipped = strHiveOpenSkipped & trim(Icase(strUserName)) & "|"
    End if
End if
End if
Next

```

'* Get Document and Settings Directory Location from the Registry

```

strKeyPath = "SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList"
objReg.GetStringValue HKEY_LOCAL_MACHINE, strKeyPath, "ProfilesDirectory", strDocAndSettingsLocation
WriteLog ""
WriteLog "Documents and Settings Path: " & strDocAndSettingsLocation
WriteLog ""
WriteLog "Checking For Orphaned Profile Directories"
WriteLog "-----"
Set objFolder = objFSO.GetFolder(strDocAndSettingsLocation)
Set colSubfolders = objFolder.Subfolders

```

```

*****
'* Parse through the directory a check For orphaned profile folders and Delete
For Each objSubfolder in colSubfolders
    strUserName = ICase(Right(objSubfolder.Path, Len(objSubfolder.Path) - InStrRev(objSubfolder.Path, "\")))
    dwProfileExclusion = Instr(strProfileExclusionList, "|" & trim(ICase(strUserName)) & "|")
    dwHiveOpenExclusion = Instr(strHiveOpenSkipped, "|" & trim(ICase(strUserName)) & "|")
    If (dwProfileExclusion = 0) and (dwHiveOpenExclusion = 0) then
        WriteLog "Deleting Orphaned Profile Directory: " & objSubfolder.Path
        If flgAllowExecute then objFSO.DeleteFolder(objSubfolder.Path), DeleteReadOnly
    Else
        If dwHiveOpenExclusion then
            WriteLog "Hive Loaded -- Skipped Delete: " & objSubfolder.Path
        End if
        If dwProfileExclusion then
            WriteLog "Profile Excluded -- Skipped Delete: " & objSubfolder.Path
        End if
    End if
End if
Next

```

```

WriteFooter
If flgLogFile then objLogFile.Close
objReg = Nothing
objFSO = Nothing
objArgs = Nothing

```

```

*****
'* Deletes All Subkeys and Values within a Given Registry Key
Sub DeleteKey(dwHiveType, strDeleteKeyPath)
    Dim dwReturn, arrDeleteSubKeys, strDeleteSubKey
    dwReturn = objReg.EnumKey(dwHiveType, strDeleteKeyPath, arrDeleteSubKeys)
    If (dwReturn = 0) And IsArray(arrDeleteSubKeys) Then
        For Each strDeleteSubKey In arrDeleteSubKeys
            DeleteKey dwHiveType, strDeleteKeyPath & "\" & strDeleteSubKey
        Next
    End if
    If flgAllowExecute then objReg.DeleteKey dwHiveType, strDeleteKeyPath
    If flgVerboseLog then WriteLog " Deleting: " & strDeleteKeyPath
End Sub

```

```

*****
'* Test a List of Wildcard Items Against a Value
Function TestCase(strTestCase, strWildCardTests)
    TestCase = 0

    dwAllItems = 0
    dwSoftLeft = 0
    dwSoftRight = 0

    arrWildCards = split(strWildCardTests, "|")

    For each strWildCard in arrWildCards
        If strWildCard = "*" then
            dwAllItems = 1

```

```

Else
  If left(strWildcard, 1) = "*" then
    dwSoftLeft = 1
    strWildcard = right(strWildcard, len(strWildcard) - 1)
  End if
  If right(strWildcard, 1) = "*" then
    dwSoftRight = 1
    strWildcard = left(strWildcard, len(strWildcard) - 1)
  End if
End if
If strWildcard = "" then dwAllItems = 1

If dwAllItems then
  TestCase = 1
Else
  dwLeftOk = 0
  dwRightOk = 0
  dwOffSet = instr(strTestCase, strWildcard)

  If dwOffSet then
    If (dwSoftLeft = 0) and (dwOffSet = 1) then dwLeftOk = 1
    If dwSoftLeft then dwLeftOk = 1
  End if
  dwRightOffSet = len(strTestCase) - dwOffSet - len(strWildcard) + 1
  If dwRightOffSet = 0 then dwRightOk = 1
  If dwRightOffSet > 0 and dwSoftRight = 1 then dwRightOk = 1
  If dwLeftOk and dwRightOk then TestCase = 1
End if
Next
End Function

*****
'* Test Profile Date Against the Current Date. Then Compare Against out Value
Function TestDateExclusion(dateTestCase, dateTestCurrentTime, dwTestNumDays)
  TestDateExclusion = 0
  dwNumDays = DateDiff("d", dateTestCase, dateTestCurrentTime)
  If dwNumDays <= dwTestNumDays then TestDateExclusion = 1
  If flgVerboseLog then WriteLog " Profile Age : " & dwNumDays & " Days"
End Function

*****
'* Log Header
Sub WriteHeader
  WriteLog "---"
  WriteLog "-- Profile Deletion Script Executed: " & dateCurrentTime
  WriteLog "---"
  WriteLog ""
End Sub

*****
'* Log Footer
Sub WriteFooter
  WriteLog ""
  WriteLog "---"
  WriteLog "-- Profile Deletion Script Completed."
  WriteLog "---"
End Sub

*****
'* Write String to Log File
Sub WriteLog(strString)
  If flgLogFile then objLogFile.Writeline strString
  If flgWriteConsole then wscript.echo strString
End Sub

```