DIGIBRANDING SIGNATURE SCRIPT Stepped Out

Here I walked the script and added some info on setup variables.

Pay Attention to notes and items in RED

START

'*** Declare some constants

'On the next line edit the path to the root folder on the server where all the signatures are stored. It is
‘assumed that there is a subfolder under this root folder for each person's signature. Edit as needed for your organization or set a specific available share location

‘ Examples also could be “**\\%logonserver%\netlogon\Signatures”** or

‘ “\\CORPDC1\NETLOGON\Signatures”

‘Be sure to leave quotes

Const SRC_PATH = “\\server1\shared\signatures “

'On the next line edit the name of the script as desired

Const SCRIPT_NAME = "Install Signature"

'*** Declare some variable

Dim objShl, objFSO, objSrc, objTgt, objFil, objFld, strSrc, strTgt, strUsr, strKey, arrKey

'*** Instantiate some variables

Set objShl = CreateObject("WScript.Shell")

Set objFSO = CreateObject("Scripting.FileSystemObject")

'*** Get the current user's name

strUsr = objShl.ExpandEnvironmentStrings("%USERNAME%")

```
'*** Set the source folder path
strSrc = SRC_PATH & "\" & strUsr

'*** Set the target folder path
strTgt = objShl.ExpandEnvironmentStrings("%APPDATA%") & "\Microsoft\Signatures"

'*** Copy the signature files from the server to the local computer for the current user if there is a signature folder for the current user
If objFSO.FolderExists(strSrc) Then
   Set objSrc = objFSO.GetFolder(strSrc)
   Set objTgt = objFSO.GetFolder(strTgt)
   For Each objFil In objSrc.Files
      objFSO.CopyFile objFil.Path, objTgt.Path & "\", True
   Next
   For Each objFld In objSrc.SubFolders
      objFSO.CopyFolder objFld.Path, objTgt.Path & "\", True
   Next

   '*** Set the default signature(s) in the registry
   'First handle the new signature setting
   strKey = GetAccountsWithSignatures("New Signature")
   If Len(strKey) > 0 Then
      arrKey = Split(strKey, "|")
      For Each strKey In arrKey
         WriteValueToRegistry strKey, "New Signature", "digitechbranding-signature"
```

```
    Next
  End If

  'Next handle the reply-forward signature setting
  strKey = GetAccountsWithSignatures("Reply-Forward Signature")
  If Len(strKey) > 0 Then
    arrKey = Split(strKey, "|")
    For Each strKey In arrKey
      WriteValueToRegistry strKey, "Reply-Forward Signature", "digitechbranding-signature"
    Next
  End If
End If

'*** Clean-up
Set objShl = Nothing
Set objFSO = Nothing
Set objSrc = Nothing
Set objTgt = Nothing
Set objFil = Nothing
Set objFld = Nothing

'End processing
WScript.Quit

Function GetAccountsWithSignatures(strType)
```

```
  Const HKEY_CURRENT_USER = &H80000001
  Const KEY_PATH = "Software\Microsoft\Windows NT\CurrentVersion\Windows Messaging Subsystem\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676"
  Dim objReg, objShl, arrProfileKeys, varSubKey, varName, varKey, strSubKeyPath
  Set objShl = CreateObject("Wscript.Shell")
  Set objReg = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\default:StdRegProv")
  objReg.EnumKey HKEY_CURRENT_USER, KEY_PATH, arrProfileKeys
  For Each varSubKey In arrProfileKeys
    'strSubKeyPath = "HKCU\" & KEY_PATH & "\" & varSubKey & "\" & strType
    strSubKeyPath = KEY_PATH & "\" & varSubKey
    strName = ""
    varName = ""
    On Error Resume Next
    varName = objShl.RegRead("HKCU\" & strSubKeyPath & "\" & strType)
    On Error GoTo 0
    If IsArray(varName) Then GetAccountsWithSignatures = GetAccountsWithSignatures & strSubKeyPath & "|"
  Next
  If Len(GetAccountsWithSignatures) > 0 Then GetAccountsWithSignatures = Left(GetAccountsWithSignatures, Len(GetAccountsWithSignatures) - 1)
  Set objReg = Nothing
  Set objShl = Nothing
End Function


'* Purpose: Convert any ASCII string to an unicode array/string (VarType
'*         = 8204). A unicode array/string is where each ASCII character
'*         is represented by two values (the ascii char value, and 0).
'*          The unicode "string" also ends in two zeros (0).
'*
'* Input:   strAny = The string to convert to the unicode array
'*
'* Return:  An array of unicode character values (VarType = 8204).
'*
'* Example: ASCII2UnicodeArray("IT") = Array(73, 0, 84, 0, 0, 0)
'*          where 73,00=I and 84,00=T and 0,0 are the ending zeros.
'*
'* Notes: This procedure is can be used with the SetBinaryValue Method
'*         of the StdRegProv Class in WMI write to the Registry.
'*
'********************************************************************
Function ASCII2UnicodeArray(ByVal strAny)
  'Version: 1.0 2007-01-26
  Dim iIndex, iPos
  ReDim aryBytes(Len(strAny) * 2 + 1)
  iIndex = -1
  For iPos = 1 To Len(strAny)
    iIndex = iIndex + 1
    aryBytes(iIndex) = Asc(Mid(strAny, iPos, 1))
    ' add a 0 after each letter
    iIndex = iIndex + 1
    aryBytes(iIndex) = 0
  Next 'iPos
  ' add two closing 0's
  iIndex = iIndex + 1
  aryBytes(iIndex) = 0
  iIndex = iIndex + 1
  aryBytes(iIndex) = 0
  ASCII2UnicodeArray = aryBytes
End Function 'ASCII2UnicodeArray


Sub WriteValueToRegistry(strPath, strName, strValue)
    Const HKEY_CURRENT_USER = &H80000001
    Dim objRegistry, uBinary, varReturn
    Set objRegistry = GetObject("Winmgmts:root\default:StdRegProv")
    uBinary = ASCII2UnicodeArray(strValue)
    varReturn = objRegistry.SetBinaryValue(HKEY_CURRENT_USER, strPath, strName, uBinary)
    If (varReturn <> 0) Or (Err.Number <> 0) Then
        MsgBox "There was an error writing the signature name to the registry", vbCritical + vbOKOnly, SCRIPT_NAME
    End If
    Set objRegistry = Nothing
End Sub
```

END