

Retrieve the shortcut properties from all the shortcuts in the Start Menu

As a packager it is needed to retrieve the shortcut properties. What executable is used in each shortcut.

On [MS Forums](#) I started a thread regarding this one. This is the final result:

```
'=====
=====
'
' VBScript Source File -- Created with SAPIEN Technologies
PrimalScript 4.0
'
' NAME: RetrieveAllShortCutProperties_v006.vbs
'
' AUTHOR: Willem-Jan
' DATE   : 04-09-2009
'
' COMMENT:
' This script is used to inventory all the shortcuts in a
given folder and
' its subfolders.
'
' Version-history:
' v.0.0.1:
'   Initial version
'
' v 0.0.2:
'   Also output to a Word Document
'
' v.0.0.3:
'   Creates only a Word Document.
'
```

```
' v.0.0.4:
'   Also makes a print screen from a shortcut.
'   Does only work on XP ENG
'
' v.0.0.5:
'   Correction Tom Lavedas added.
'
' v.0.0.6:
'   Improved Windows 7 support
'
```

```
'=====
=====
```

Option Explicit

```
'=====
=====
'Declare the variables
'=====
=====
```

```
Dim objApplication
Dim objFolder
Dim objStartFolder
Dim objFSO
Dim objFile
dim objShell
Dim objFolderItem
dim objApplicationLink
Dim objWord
Dim objExplorer
Dim objWordBasic
Dim strWordOutputFile
dim objOutputFile
Dim subFolder
dim strFolder
Dim colFiles
Dim cntLNKFiles
Dim strShortcut
Dim objRange
Dim objTable
```

```

Dim objDoc
Dim objSelection
dim cntTables
Dim sLnkPath
Dim sLnkName
Dim strAction
Dim strArguments
Dim strLogFile
Dim strEntry
Dim oFile

Dim arrShortcut
Dim strShortcutName
Dim strShortcutFolder

Dim strReadyMessage

'=====
=====
'Set the variables
'=====
=====

Set    objFSO                                =
CreateObject("Scripting.FileSystemObject")
Set objApplication    = CreateObject("Shell.Application")
Set objShell          = CreateObject("WScript.Shell")
Set objWord           = CreateObject("Word.Application")
Set objWordBasic     = CreateObject("Word.Basic")
Set    objExplorer   =                                =
createobject("internetexplorer.application")
cntLNKFiles          = 0
cntTables            = 1

Const END_OF_STORY          = 6
Const MOVE_SELECTION       = 0
Const LOGGING               = True
Const wdLineStyleDot       = 2
Const wdLineStyleDashDotDot = 6

```

```
Const wdLineStyleDashDot      = 5
Const wdLineStyleNone         = 0
Const wdBorderBottom          = -3
Const wdBorderHorizontal      = -5
Const wdBorderLeft            = -2
Const wdBorderRight           = -4
Const wdBorderTop             = -1
Const wdBorderVertical        = -6
Const wdBorderDiagonalDown    = -7
Const wdBorderDiagonalUp     = -8
```

```
strLogFile = "C:\TEMP\Discover_shortcuts.log"
```

```
Set objFolder          = objApplication.BrowseForFolder (0,
"Select the folder to search for .lnk files:", (0))
If objFolder Is Nothing Then
    Wscript.Quit
    Else
    Set objFolderItem = objFolder.Self
    objStartFolder   = objFolderItem.Path
End If
```

```
'=====
=====
'If the selected folder does not end with a '\' add one.
'=====
=====
```

```
If Right(objStartFolder,1)<>"\" Then
    objStartFolder = objStartFolder & "\"
End If
```

```
'=====
=====
'Define the outfile
'=====
=====
```

```

strWordOutputFile      =      objStartFolder      &
"_resultsfile_LNK_files.doc"

'=====
'Delete a previous version (if exists)
'=====

If objFS0.FileExists(strWordOutputFile) Then
    objFS0.DeleteFile(strWordOutputFile)
end If

'=====
'Create C:\TEMP (if not exists)
'=====

If Not objFS0.FolderExists("C:\TEMP") Then
    objFS0.CreateFolder("C:\TEMP")
end If

'=====
'Delete a previous logfile (if exists)
'=====

If LOGGING Then
    If objFS0.FileExists(strLogFile) Then
        objFS0.DeleteFile(strLogFile)
    end If
    Set objOutputFile      = objFS0.CreateTextFile
(strLogFile)
End If

'=====
'Create the Word document
'=====

```

```

objWord.Visible      = False
Set objDoc           = objWord.Documents.Add()
Set objSelection     = objWord.Selection
If LOGGING Then
    WriteToLogFile("Word document generated in the background")
End If

'=====
'Al dan niet tonen van mededelingen op het scherm.
'=====

objExplorer.Navigate "about:blank"
objExplorer.ToolBar      = 0
objExplorer.StatusBar   = 0
objExplorer.Width       = 600
objExplorer.Height      = 200
objExplorer.Left        = 100
objExplorer.Top         = 100
Do While (objExplorer.Busy)
    Wscript.Sleep 200
Loop
objExplorer.Visible     = 1

'=====
'Look in the root of the given foldername.
'=====

Set objFolder = objFS0.GetFolder(objStartFolder)
Set colFiles = objFolder.Files
For Each objFile in colFiles
    if lcase(right(objFile,3)) = "lnk" Then
        cntLNKFiles = cntLNKFiles + 1
        If LOGGING Then
            WriteToLogFile("LNK file found and processing: " &
objFile)
        End If
        Readshortcut(objFile)
    end if
end for

```

```

        end if
Next

'=====
=====
'And call the subfolders.
'=====
=====

ShowSubFolders objFSO.GetFolder(objStartFolder)

objDoc.SaveAs(strWordOutputFile)
objWord.Quit
objExplorer.Quit

strReadyMessage = "The script has finished." & vbCrlf & "See
the resultsfile (doc): " & strWordOutputFile & "." & vbCrlf
strReadyMessage = strReadyMessage & "There are " & cntLNKFiles
& " shortcuts found in the folder " & objStartFolder & " and
its subfolders."
If LOGGING then
    strReadyMessage = strReadyMessage & vbCrlf & "The logfile
is: " & strLogFile & "."
end if

wscript.echo strReadyMessage

'=====
=====
'Procedure: Readshortcut(strShortcut)
'Target: request shortcut properties
'Usage:
'    Readshortcut(strShortcut) where strShortcut is the lnk
file name.
'=====
=====

Sub Readshortcut(strShortcut)

```

```
set objApplicationLink = objShell.CreateShortcut(strShortcut)
' Execute the createshortcut method, which also retrieves an
existing shortcut
```

```
ShowProgress("Processing file: " & strShortcut)
```

```
arrShortcut      = split(strShortcut, "\")
strShortcutName  = arrShortcut(ubound(arrShortcut))
strShortcutFolder = left(strShortcut, len(strShortcut) -
len(strShortcutName))
```

```
If LOGGING Then
    WriteToLogFile("      Value 'strShortcutName': " &
strShortcutName)
    WriteToLogFile("      Value 'strShortcutFolder': " &
strShortcutFolder)
End If
```

```
objSelection.TypeParagraph()
Set objRange = objSelection.Range
objDoc.Tables.Add objRange, 7, 2
Set objTable = objDoc.Tables(cntTables)
objWord.Selection.Style = "Tabelraster"
```

```
With objDoc.Tables(cntTables)
    .Borders(wdBorderLeft).LineStyle      = wdLineStyleDot
    .Borders(wdBorderRight).LineStyle     = wdLineStyleDot
    .Borders(wdBorderTop).LineStyle       = wdLineStyleDot
    .Borders(wdBorderBottom).LineStyle    = wdLineStyleDot
    .Borders(wdBorderVertical).LineStyle  = wdLineStyleDot
    .Borders(wdBorderHorizontal).LineStyle = wdLineStyleDot
    .Borders(wdBorderDiagonalDown).LineStyle = wdLineStyleNone
    .Borders(wdBorderDiagonalUp).LineStyle = wdLineStyleNone
    .Borders.Shadow = False
End With
```

```
objTable.Cell(1, 2).Select
Call getSCPropImage (objFile)
```



```

objTable.Cell(2, 1).Range.Text = "Shortcutname:"
objTable.Cell(2, 2).Range.Text = strShortcut

objTable.Cell(3, 1).Range.Text = "Targetpath:"
objTable.Cell(3, 2).Range.Text =
objApplicationLink.TargetPath & " " &
objApplicationLink.Arguments

objTable.Cell(4, 1).Range.Text = "Working directory:"
objTable.Cell(4, 2).Range.Text =
objApplicationLink.WorkingDirectory

objTable.Cell(5, 1).Range.Text = "Hotkey:"
objTable.Cell(5, 2).Range.Text = objApplicationLink.Hotkey

objTable.Cell(6, 1).Range.Text = "Description:"
objTable.Cell(6, 2).Range.Text =
objApplicationLink.Description

objTable.Cell(7, 1).Range.Text = "Icon location:"
objTable.Cell(7, 2).Range.Text =
objApplicationLink.IconLocation

objTable.Columns.Item(1).Width = objWord.InchesToPoints(2)
objTable.Columns.Item(2).Width = objWord.InchesToPoints(5)

objSelection.EndKey END_OF_STORY
objSelection.TypeParagraph()

cntTables = cntTables + 1

```

End Sub

```

'=====
'
'Procedure: ShowSubFolders(strFolder)
'Target: Search for lnk files in this folder and its
subfolders
'Usage:
' ShowSubFolders(strFolder) where strFolder is folder name.
'=====

```

```

=====
Sub ShowSubFolders(strFolder)
  For Each Subfolder in strFolder.SubFolders
    Set objFolder = objFS0.GetFolder(Subfolder.Path)
    Set colFiles = objFolder.Files
    For Each objFile in colFiles
      if lcase(right(objFile,3)) = ".lnk" Then
        cntLNKFiles = cntLNKFiles + 1
        Readshortcut(objFile)
      end if
    Next
    ShowSubFolders subFolder 'Use recursion to investigate
the underlying levels.
  Next
End Sub

```

```

' =====
' =====
' Procedure: getSCPropImage(oFile)
' Target: Collect an image of the shortcut's Properties
' Usage:
'   getSCPropImage oFile
'
'   where oFile is a file object pointing to the shortcut lnk
file
'
' (C) Tom Lavedas
' =====
' =====

```

```

Sub getSCPropImage(oFile)
dim sName : sName = replace(lcase(oFile.Name), ".lnk", "")
Dim sPath : sPath = objFS0.GetParentFolderName(oFile)
Dim oFldr : set oFldr = objApplication.Namespace(sPath)
Dim oTemp : set oTemp = oFldr.ParseName(oFile.Name)
Dim objVerb, colVerbs
If Right(sPath,1) <> "\" Then
  sPath = sPath & "\"
End If

```

```

If LOGGING Then
    WriteToLogFile(" Running the procedure getSCPropImage")
    WriteToLogFile("    Value 'sPath': " & sPath)
    WriteToLogFile("    Value 'sName': " & sName)
    WriteToLogFile("        Result fnFolderItemVerbsVB: " &
fnFolderItemVerbsVB)
End If

dim sVerbName : sVerbName = fnFolderItemVerbsVB

If sVerbName <> "" then
    oTemp.InvokeVerb(sVerbName)
    If LOGGING Then
        WriteToLogFile("    The verb '" & sVerbName & "' has been
found")
    end if
else
    If LOGGING Then
        WriteToLogFile("    There is an error - verb '" &
sVerbName & "' not found")
    end if
    exit sub
end if

sVerbName = Replace(sVerbName, "&", "")

Dim n

If LOGGING Then
    WriteToLogFile("    Processing: " & sName)
end if
n = 0

Set colVerbs = oTemp.Verbs
For Each objVerb in colVerbs
    If Replace(objVerb.name, "&", "") = sVerbName Then
        objVerb.DoIt
        if LOGGING Then
            WriteToLogFile("    The command '" & sVerbName & "'
has been executed.")
        end if
    end if
end if

```

```

        do until objShell.AppActivate(sName) or (n > 40) ' wait
2 seconds, only
            wsh.sleep 50
            n = n + 1
        loop
    end if
Next

if objShell.AppActivate(sName) then
    wscript.sleep 2000 ' Wait 2 seconds so the screen is
fully shown. Sometimes there is a delay when showing Windows 7
shortcut properties
    if LOGGING Then
        WriteToFile(" " & sName & " has been found.")
    end if
    objWordBasic.SendKeys "{PrtSc}", True ' Other
implementations do not support this keystroke
    wscript.sleep 100
    objShell.sendkeys "{F4}", True ' Closes Property
dialog
    If LOGGING Then
        WriteToFile(" Details copied to Word (Office)
clipboard")
    end If
    objSelection.Collapse 1
    On Error Resume Next
    objSelection.Paste
    if LOGGING then
        if Err.Number = 0 Then
            WriteToFile(" Details pasted into Word")
        else
            WriteToFile(" Details NOT pasted into Word due
to a Office Clipboard error.")
        end if
    End If
    On Error Goto 0
else
    If LOGGING Then
        WriteToFile(" There is an error while showing the
properties for screen '" & sName & "'")
        WriteToFile(" The procedure getSCPproperties will

```

```

be quit.")
    WriteToLogFile("-----
-----" & vbCrLf)
    End If
    Exit Sub
end if

do until NOT objShell.AppActivate(sName): wscript.sleep 50
:loop

If LOGGING Then
    WriteToLogFile("-----
-----" & vbCrLf)
End If
end Sub

' =====
=====
'Procedure: ShowProgress
'Doel: Show a message on the screen.
'Usage:
' ShowProgress(string)
'where 'string' contains the text to be shown.
' =====
=====

Sub ShowProgress(strAction)
    objExplorer.Document.Body.InnerHTML = strAction
End Sub

' =====
=====
'Procedure: WriteToLogFile
'Usage: Write a message in the log file
'Use:
' ShowProgress(stringwaarde)
'where 'string' contains the text to be written to the log
file.
' =====
=====

```

=====

```
Sub WriteToLogFile(strEntry)
    objOutputFile.WriteLine (Date & " - " & Time & " -> "&
strEntry)
End Sub
```

'=====

=====

```
'Function: fnFolderItemVerbsVB()
'Usage: find the string which has to be executed to get the
properties.
'Use:
'  fnFolderItemVerbsVB()
'
```

'=====

=====

```
function fnFolderItemVerbsVB()
    dim objShell, s
    set objShell = CreateObject("Shell.Application")
    if (not objShell is nothing) then
        dim objFolder2
        Const ssfWINDOWS = 36
        set objFolder2 = objShell.NameSpace(ssfWINDOWS)
        if (not objFolder2 is nothing) then
            dim objFolderItem
            set objFolderItem = objFolder2.Self
            if (not objFolderItem is nothing) then
                dim objItemVerbs
                set objItemVerbs = objFolderItem.Verbs
                if (not objItemVerbs is nothing) then
                    s = objItemVerbs.item(objItemVerbs.count - 1)
                end if
            end if
        end if
    end if
    fnFolderItemVerbsVB = s
```

```
end function
```