

# MakeUnzipFolder.vbs

```

1  '*****
2  'Name:          MakeUnzipFolder.vbs          *
3  'Beschreibung:  Fügt die übergebenen Dateien / Ordner einem          *
4  '              Komprimierten Ordner mit der Erweiterung "unzip_",      *
5  '              über die Windowseigene Zip-Funktion, hinzu.          *
6  '              *
7  'Parameter:     Datei(en) und / oder Ordner mit Pfad                  *
8  'Version:       1.0                                                    *
9  'Datum:         2008-08-21                                             *
10 'Autor:         RR                                                    *
11 '*****
12 Option Explicit
13 ' Schalter für Erfolgsmeldung (Fehler werden immer angezeigt)
14 Const Message = True
15
16 Main Message
17
18 Sub Main(bolMessage)
19     Dim arrArg                ' Array für übergebene Argumente
20     Dim strDst                ' Variable für Zielfeld (Zip-Datei)
21     Dim bolResult             ' Variable für Ergebnis der Komprimierung
22
23     ' Initialisierung der Ergebnisvariablen.
24     bolResult = False
25
26     ' Übergabe der Argumente an Array.
27     Set arrArg = WScript.Arguments
28     ' Zum Debuggen, Dateien / Ordner müssen vorhanden sein! (Set arrArg) auskommentieren
29     'arrArg = Array("D:\Testordner1")
30     'arrArg = Array("D:\Testordner1", "D:\Testordner2")
31     'arrArg = Array("D:\Testdatei1", "D:\Testdatei2.txt")
32     'arrArg = Array("D:\Testordner1", "D:\Testdatei1.txt")
33
34     ' Erstellen des Zielpfades anhand der ersten übergebenen Datei / Ordner
35     strDst = getDestination(arrArg(0))
36     ' Überprüft ob eine gültiger Zielpfad erstellt wurde.
37     If strDst <> "" Then
38         ' Erstellt die Ziel-Zip-Datei
39         CreateZipFile strDst
40         If IsFile(strDst) Then
41             ' Dateien bzw. Ordner komprimieren
42             ZipFolder arrArg, strDst
43             ' Meldung wenn Komprimierung abgeschlossen ist.
44             If bolMessage Then
45                 CreateObject("WScript.Shell").SendKeys "{F10}{ESC}", True
46                 MsgBox "Die Zip-Datei wurde erfolgreich erstellt.", 0, "Komprimierung"
47             End If
48         Else
49             CreateObject("WScript.Shell").SendKeys "{F10}{ESC}", True
50             MsgBox "Fehler beim erstellen des Zip-Ordners!", 16, "Komprimierung"
51         End If
52     Else
53         CreateObject("WScript.Shell").SendKeys "{F10}{ESC}", True
54         MsgBox "Die Datei" & vbCrLf & _
55             arrArg(0) & vbCrLf & _
56             "kann nicht Komprimiert werden.", 16, "Komprimierung"
57     End If
58     Set arrArg = Nothing
59 End Sub
60
61 ' Erstellt aus dem Quellpfad und Dateinamen, den Zielpfad und Dateinamen mit "unzip_"
62 Function getDestination(strSource)
63     Dim objFso
64     Dim intNameLen
65     Dim strDestination, strPath
66     Const strBefore = "unzip_"      ' Erweiterung für Unzipper
67     Const strExtension = ".zip"     ' Dateiendung ".zip"
68
69     Set objFso = CreateObject("scripting.filesystemobject")
70
71     If IsFolder(strSource) Then
72         Dim objFld

```

```

73     Set objFld = objFso.GetFolder(strSource)
74     intNameLen = Len(objFld.Name)
75     strPath = Left(objFld.Path, Len(objFld.Path) - intNameLen)
76     strDestination = strPath & strBefore & objFld.Name & strExtension
77     getDestination = strDestination
78     Set objFld = Nothing
79 Else
80     Dim objFile, strName, arrName
81     Set objFile = objFso.GetFile(strSource)
82     intNameLen = Len(objFile.Name)
83     arrName = Split(objFile.Name, ".", -1)
84     strPath = Left(objFile.Path, Len(objFile.Path) - intNameLen)
85     strDestination = strPath & strBefore & arrName(0) & strExtension
86     getDestination = strDestination
87     Set objFile = Nothing
88 End If
89
90 Set objFso = Nothing
91 End Function
92
93 ' Kopiert den angegebenen Pfad in den Zip-Ordner
94 Sub ZipFolder(arrSource, strDestination)
95     Dim objApp, objFso, objSrc
96     Dim strSource, strParent, strFilter
97     ' Konstanten für Filter
98     Const SHCONTF_FOLDERS = 32
99     Const SHCONTF_NONFOLDERS = 64
100    Const SHCONTF_INCLUDEHIDDEN = 128
101
102    Set objApp = CreateObject("Shell.Application")
103    Set objFso = CreateObject("Scripting.FileSystemObject")
104
105    ' Alle übergebene Dateien / Ordner zur Zip-Datei hinzufügen
106    For Each strSource In arrSource
107        If IsFolder(strSource) Then
108            Dim objFld
109            Set objFld = objFso.GetFolder(strSource)
110            ' Übergeordneten Pfad ermitteln
111            strParent = objFld.ParentFolder.Path
112            Set objSrc = objApp.Namespace(strParent).Items
113            ' Filter auf aktuellen Ordnernamen beschränken
114            strFilter = objFld.Name
115            ' Filter "Ordner" zuweisen
116            objSrc.Filter SHCONTF_FOLDERS, strFilter
117            Set objFld = Nothing
118        Else
119            Dim objFile
120            Set objFile = objFso.GetFile(strSource)
121            ' Übergeordneten Pfad ermitteln
122            strParent = objFile.ParentFolder.Path
123            Set objSrc = objApp.Namespace(strParent).Items
124            ' Filter auf aktuelle Datei beschränken
125            strFilter = objFile.Name
126            ' Filter "Dateien" oder "Versteckt" zuweisen
127            objSrc.Filter SHCONTF_NONFOLDERS Or SHCONTF_INCLUDEHIDDEN, strFilter
128            Set objFile = Nothing
129        End If
130        ' Ordner / Dateien der Zip-Datei hinzufügen
131        objApp.Namespace(strDestination).CopyHere objSrc
132        CheckBusy(300)
133    Next
134    CheckBusy(1000)
135    Set objFso = Nothing
136    Set objSrc = Nothing
137    Set objApp = Nothing
138 End Sub
139
140 ' Überprüft ob Komprimierungsprozess abgeschlossen ist. (intTime in Millisekunden)
141 Sub CheckBusy(intTime)
142     Dim intCounter
143
144     With GetObject("winmgmts:Win32_PerfFormattedData_PerfProc_Process.Name='wscript'")
145         Do While intCounter < 2

```

```

146         .Refresh_: WScript.Sleep intTime
147         If .PercentProcessorTime = 0 Then
148             intCounter = intCounter + 1
149         Else
150             intCounter = 0
151         End If
152     Loop
153 End With
154 End Sub
155
156 ' Erstellt einen leeren Zip-Ordner mit entsprechendem Header
157 Sub CreateZipFile(strFile)
158     Dim objFso
159     Set objFso = CreateObject("Scripting.FileSystemObject")
160
161     objFso.CreateTextFile(strFile, True).Write _
162         Chr(80) & Chr(75) & Chr(5) & Chr(6) & String(18, Chr(0))
163
164     Set objFso = Nothing
165 End Sub
166
167 ' Überprüft ob übergebener Pfad ein Ordner ist
168 Function IsFolder(strPath)
169     Dim objFso
170     Set objFso = CreateObject("scripting.filesystemobject")
171
172     If objFso.FolderExists(strPath) Then
173         IsFolder = True
174     Else
175         IsFolder = False
176     End If
177
178     Set objFso = Nothing
179 End Function
180
181 ' Überprüft ob übergebener Pfad eine Datei ist
182 Function IsFile(strPath)
183     Dim objFso
184     Set objFso = CreateObject("scripting.filesystemobject")
185
186     If objFso.FileExists(strPath) Then
187         IsFile = True
188     Else
189         IsFile = False
190     End If
191
192     Set objFso = Nothing
193 End Function

```

1 Name: MakeUnzipFolder.vbs  
2  
3 Lizenz: Dieses Skript steht zur freien Verfügung, da ich selbst  
4 die erforderlichen Informationen und Routinen aus freien  
5 Quellen bezogen habe.  
6  
7 Plattform: Getestet unter windows XP und Vista.  
8  
9 Sprache: VB-Script.  
10  
11 IDE: VbsEdit 3.4.1.6 <http://www.vbsedit.com>  
12  
13 Dateien: MakeUnzipFolder.vbs : Das Script  
14 Readme.txt : Beschreibung  
15 MakeUnzipFolder.pdf : Source + Readme  
16  
17 Disclaimer: Die Anwendung des Scripts erfolgt auf eigene Gefahr.  
18 Da das Script zur freien Verfügung steht,  
19 kann ich keinen Support geben.  
20 Für konstruktive Kritik sowie Anregungen oder  
21 Verbesserungsvorschläge bin ich immer offen.  
22  
23 Voraussetzung: Die windowseigene Zip-Funktion muss aktiviert und  
24 funktionstüchtig sein.  
25  
26 Beschreibung: Das Skript wurde für das Plugin "OnTheFlyUnzipper" des  
27 windows-Home-Servers konzipiert.  
28 Dieses Plugin überwacht angegebene Ordner ob eine  
29 Zip-Datei mit der Erweiterung "unzip\_<Dateiname>.zip"  
30 hinzugefügt wurde und Dekomprimiert diese automatisch.  
31  
32 Um die Erstellung einer Zip-Datei mit der Erweiterung  
33 "unzip\_" zu vereinfachen habe ich dieses Skript  
34 programmiert.  
35  
36 Das Skript kopiert die übergebenen Datei(en) / Ordner  
37 mit der windowseigenen Zip-Funktionalität in einen  
38 Zip-Komprimierten Ordner mit der Erweiterung "unzip\_".  
39 Hierzu wird der Datei- bzw. Ordner-Name der ersten Datei  
40 für den neuen Zip-Ordner verwendet.  
41  
42 Für eine effektive Benutzung, empfiehlt es sich, eine  
43 Verknüpfung im "Kontextmenü - Senden An" anzulegen.  
44  
45 windows-XP:  
46 C:\Dokumente und Einstellungen\USERNAME\SendTo  
47  
48 windows-Vista:  
49 C:\Users\USERNAME\AppData\Roaming\Microsoft\Windows\SendTo  
50  
51 Technisches: Da die "zipfldr.dll" welche für die windowseigene  
52 Zip-Funktion zuständig ist, nicht direkt angesprochen  
53 werden kann, muss der Umweg über das "Shell-Application"-  
54 Objekt benutzt werden.  
55  
56 Zuerst wird mit der Funktion "getDestination(strSource)"  
57 der neue Pfad mit Dateiname aus dem ersten übergebenen  
58 Parameter erstellt.  
59  
60 Wurde der Pfad erfolgreich erstellt, wird mit der  
61 Prozedur "CreateZipFile(strFile)" eine leere Zip-Datei,  
62 mit dem erforderlichen Header, im selben Pfad wie die  
63 Quelldatei erstellt.

64  
65 Nach dem erfolgreichen erstellen der Zielfeile,  
66 werden die übergebenen Dateien bzw. Ordner in der Prozedur  
67 "ZipFolder(arrSource, strDestination)" mit Hilfe des  
68 "Shell-Application-Objektes" in diese kopiert.  
69 Da die Filter für Dateien und Ordner über das  
70 "Shell-Application-Objekt" unterschiedlich angesprochen  
71 werden, muß dieses für jede Datei bzw. Ordner durch die  
72 Funktion "IsFolder(strPath)" oder "IsFile(strPath)"  
73 überprüft werden.  
74  
75 Der Filter sorgt außerdem dafür, dass auch versteckte  
76 Dateien kopiert werden, da das "Shell-Application-Objekt"  
77 diese standardmäßig nicht berücksichtigt.  
78  
79 Über die Prozedur "CheckBusy(intTime)" wird in den  
80 angegebenen Zeitabständen überprüft, ob der  
81 Komprimierungsprozess für die aktuelle Datei bzw. den  
82 aktuellen Ordner abgeschlossen ist.  
83 Dies ist notwendig, da das Script ansonsten sofort mit  
84 der nächsten Datei bzw. dem nächsten Ordner fortfahren  
85 würde, was zu Fehlern führt.  
86  
87 Wurden alle Dateien bzw. Ordner komprimiert, erfolgt eine  
88 Erfolgsmeldung. Diese kann durch setzen der Konstante  
89 "Message" auf "False" auch unterdrückt werden.  
90  
91 Anmerkung: Die Komprimierung erfolgt etwas langsamer, als bei der  
92 original Zip-Funktion, was aber durch die sonst nötige  
93 Ubenennung der erstellten Zip-Datei nicht wesentlich ins  
94 Gewicht fällt.  
95 Ich habe den Code so gut es geht kommentiert, so dass  
96 auch Anfänger bzw. Unerfahrene Benutzer sich schnell  
97 zurechtfinden sollten und das Script den eigenen  
98 Bedürfnissen anpassen können.  
99 Das Schript soll auch Anregung geben diese Funktionalität  
100 bei Zip-Tools anderer Hersteller wie z.B. "WinZip" oder  
101 "7-Zip" zu implementieren.  
102  
103  
104  
105  
106  
107  
108  
109