

Microsoft® Windows® Script Host

by Günter Born

Windows Script Host

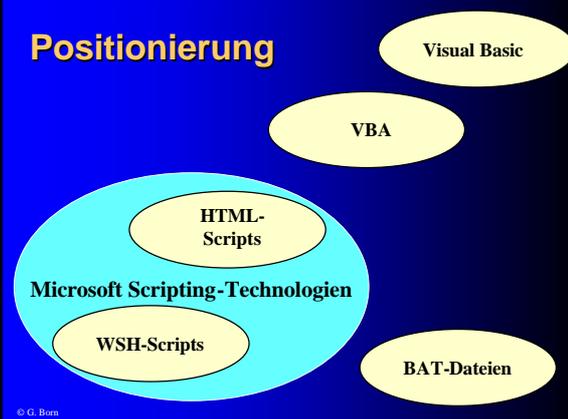
- Grundlagen
- Einsatzmöglichkeiten
- Beispiele
- Erweiterte Technologien
- Informationen

© G. Born

Windows Script Host
oder
Windows Scripting Host?
= WSH

© G. Born

Positionierung



© G. Born

WSH Was ist das?

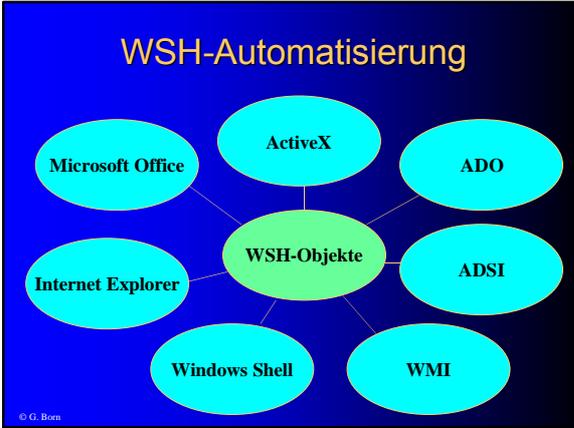
- Host für ActiveX®-Skriptmodule
- Sprachenunabhängig
- Windows-Skripttool
- Ausführbare Anwendungen
- WSH eignet sich für ...

© G. Born

Host für ActiveX®- Skriptmodule

- WSH stellt ActiveX-Schnittstellen (Objekte, Methoden und Eigenschaften) für die direkte Skriptaufführung bereit

© G. Born



WSH Was ist das?

- Host für ActiveX®-Skriptmodule.
- **Sprachenunabhängig**
- **Windows-Skripttool**
- **Ausführbare Anwendungen**
- **WSH eignet sich für ...**

© G. Born

Sprachenunabhängig

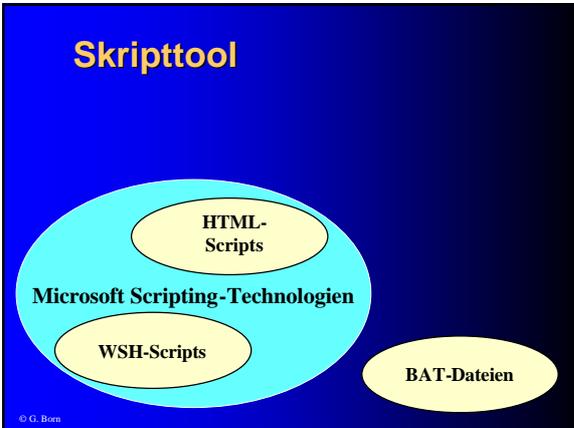
- **Microsoft Script-Engines**
 - Microsoft VBScript
 - Microsoft JScript
- **Script-Engines von Drittherstellern**
 - Perlscript, TCL, REXX, Python ...

© G. Born

WSH Was ist das?

- Host für ActiveX®-Skriptmodule.
- **Sprachenunabhängig**
- **Windows-Skripttool**
- **Ausführbare Anwendungen**
- **WSH eignet sich für ...**

© G. Born



Windows-Skripttool

- **Microsoft Windows 98 (WSH 1.0)**
- **Microsoft Windows NT 4.0 (OP 4.0)**
- **Microsoft Windows 2000 (WSH 2.0)**

WSH 2.0 nachrüstbar für

- **Microsoft Windows 95/98**
- **Microsoft Windows NT 4.0**

© G. Born

WSH Was ist das?

- Host für ActiveX®-Skriptmodule.
- Sprachenunabhängig
- Windows-Skripttool
- Ausführbare Anwendungen
- WSH eignet sich für ...

© G. Born

Ausführbare Anwendungen

- Skriptdateien .vbs, .js und .wsf
- Host WScript.exe (Windows Host)
 - Ausführung per Doppelklick
 - Über Drag & Drop
 - Über Dialogfeld Ausführen im Startmenü
- Host CScript.exe (Eingabeaufforderung)
 - Cscript.exe <optionen> <pfad>skriptname

© G. Born

Skriptaufruf

Wscript /?



© G. Born

WSH Was ist das?

- Host für ActiveX®-Skriptmodule.
- Sprachenunabhängig
- Windows-Skripttool
- Ausführbare Anwendungen
- WSH eignet sich für ...

© G. Born

WSH eignet sich für ...

- Meldungen ausgeben
- Zuordnen von Netzwerklaufrufen
- Verbinden zu Druckern
- Abrufen und Bearbeiten von Umgebungsvariablen
- Ändern von Registrierungseinträgen
- Ausführen elementarer Funktionen
 - z.B. Zugriffe auf andere COM-Objekte

© G. Born

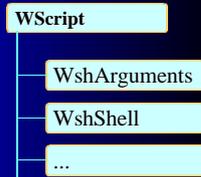
WSH-Skripte eignen sich für ...

- Allgemeine Automatisierungsaufgaben (Anmeldung, Protokollierung)
- Sicherung und Manipulation von Dateien
- Zugriffe auf Datenbank (ADO)
- Benutzerverwaltung in Windows NT/2000 Netzwerken (ADSI)
- Netzwerkweite Administration (WBEM/WMI)

© G. Born

WSH-Objektmodell

- WScript-Objekt
- WshArguments-Objekt
- WshEnvironment-Objekt
- WshNetwork-Objekt
- WshShell-Objekt
- WshShortcut-Objekt
- WshSpecialFolders-Objekt
- WshUrlShortcut-Objekt



© G. Born

VBSkript-Beispiel

```

' Anwenden der Echo-Methode
Option Explicit
Dim text
text = "Windows Script Host" ' Text definieren
WScript.Echo "Dialogfeld per", _
vbCrLf & text, vbCrLf, "anzeigen"
WScript.Quit
  
```



© G. Born

JScript-Beispiel

```

// Anwenden der Echo-Methode
var text = "Windows Script Host";
WScript.Echo ("Dialogfeld per \n" + text,
"\nanzeigen");

WScript.Quit();
  
```



© G. Born

Zugriff auf Argumente

```

' Anzeige übergebener Parameter
Option Explicit
Dim text, oArgs, I
text = "Argumente" & vbCrLf & vbCrLf
Set oArgs = Wscript.Arguments ' Objekt erzeugen
For I = 0 to oArgs.Count - 1 ' über alle Argumente
    text = text & oArgs(I) & vbCrLf ' hole Argument
Next
Wscript.Echo text ' zeige Argument per Echo
  
```



© G. Born

Anwendung ausführen

```

' Windows-Editor per Run-Methode starten
Option Explicit
DIM WshShell

Set WshShell = WScript.CreateObject ("WScript.Shell")
WshShell.Run "%Windir%\Notepad.exe", 1
WScript.Echo "Quellcode im minimierten, & _
"Editorfenster laden.,
WshShell.Run "Notepad.exe " & _
WScript.ScriptFullName, 6
WScript.Quit
  
```

© G. Born

Drucker verbinden

```

' Weist dem Drucker LPT1: einen Netzwerkdrucker zu
Set WshNetwork = WScript.CreateObject("WScript.Network")

On Error Resume Next
' Windows 9x
WshNetwork.AddPrinterConnection "LPT1", "\\ROM\HP500"

' Windows NT/2000
WshNetwork.AddWindowsPrinterConnection "\\ROM\HP500"

' Setze Standarddrucker
WshNetwork.SetDefaultPrinter "HP 500"
  
```

© G. Born

Netzlaufwerk verbinden

```
' Netzlaufwerke zuweisen & auflisten
Set WshNetwork = WScript.CreateObject("WScript.Network")
On Error Resume Next
WshNetwork.MapNetworkDrive "K:", "\\ROMC"
Set oDrives = WshNetwork.EnumNetworkDrives
For i = 0 To oDrives.Count-1 Step 2
  If oDrives(i) <> " " Then
    Text = Text + oDrives(i) + " " + oDrives(i+1) + vbCRLF
  End if
Next
WScript.Echo Text
```

© G. Born

Zugriff auf Dateien

- Erfolgt per FileSystemObject-Objekt
- In VBScript/JScript direkt enthalten
- Bietet Methoden zum:
 - Auflisten von Ordnern/Dateien
 - Kopieren, Verschieben, Löschen
 - Zugriff auf Textdateien

© G. Born

Datei manipulieren

```
' Kopiert und löscht eine Datei
Const file1 = "C:\Born\Test.txt,"
Const file2 = "C:\Born\Test1.txt"

Set fso = CreateObject("Scripting.FileSystemObject")
If (fso.FileExists(file1)) Then
  ' kopiere die Datei file1 nach file2
  Set fi = fso.GetFile(file1) ' hole Dateiobjekt
  fi.Copy file2
  WScript.Echo "Datei " & file2 & " kopiert"
  ' Datei ggf. löschen
  Set fi = fso.GetFile(file2) ' hole Dateiobjekt
  fi.Delete
  WScript.Echo "Datei " & file2 & " gelöscht"
End if
WScript.Quit
```

© G. Born

Zugriff auf Excel

```
' Zugriff auf Excel per WSH-Skript
Set objXL = WScript.CreateObject("Excel.Application")
' Jetzt für die Anzeige des Fensters sorgen
objXL.WindowState = 1 ' Normal
objXL.Height = 200 ' Höhe
objXL.Width = 400 ' Breite
objXL.Left = 100 ' X-Position
objXL.Top = 100 ' Y-Position
objXL.Visible = true ' Fenster anzeigen
objXL.Caption = Title ' Titeltext setzen
objXL.Statusbar = "Datum: " & Date ' Statusleiste
' Frage jetzt die Excel-Eigenschaften ab
WScript.Echo "Fenstertitel " & objXL.Caption
objXL.Quit ' Excel jetzt schließen
```

© G. Born

Zugriff auf Word

```
' Word per WSH-Skript ansprechen
Set objWrd = WScript.CreateObject("Word.Application")
objWrd.WindowState = 1 ' Maximiert
objWrd.Visible = true ' Fenster anzeigen
objWrd.Caption = Title ' Titeltext
objWrd.Statusbar = "Datum: " & Date ' Statustext
WScript.Echo "Fensterhöhe " & vbTab & objWrd.Height _
vbCRLF & "Fensterbreite " & vbTab & objWrd.Width
objWrd.Quit ' Word jetzt schließen
```

© G. Born

Weitere Technologien

- Datenbankzugriffe (ADO)
- Administration (ADSI)
- Systemadministration (WMI)

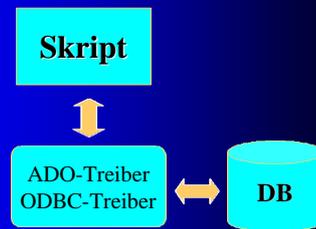
© G. Born

ActiveX Data Objects (ADO)

- ADO ist die Schnittstelle zum Zugriff auf Datenbanken
 - (ADO- oder ODBC-Treiber)
 - SQL, Access, XLS, CSV, Text etc.
- ADO-Bibliotheken
 - Datenzugriff
 - Verwaltung Datenbankstruktur

© G. Born

ADO-Zugriffe



© G. Born

ADO-Beispiel

```
' ActiveX Data Objects (ADO)-Beispiel, liest CSV-Datei
' Treiber einstellen, Wichtig: dbq=C:\ setzt Pfad auf Dummy
drv = "driver={Microsoft Text-Treiber (*.txt; *.csv)}; dbq=C:\\"
' Erzeuge Connection-Objekt
Set conn = WScript.CreateObject("ADODB.Connection")
conn.Open drv ' kein DSN, nur Treiber angeben
' hole Recordset-Objekt mit Daten, gebe Datei beim Open Aufruf an
Set rs = WScript.CreateObject("ADODB.Recordset")
rs.Open "C:\BornNamen.csv", conn, , , 2 ' adCmdTable
Do While Not rs.EOF ' alle Sätze
txt = txt & rs("Name:STRING") & vbTab & rs("ID:String") & vbCRLF
rs.MoveNext ' nächster Satz
Loop
WScript.Echo txt
conn.Close ' Verbindung schließen
WScript.Quit ' Skript beenden
```

© G. Born

Active Directory Services Interface (ADSI)

- Schnittstelle zum Zugriff auf Active Directory Services
- Ab Windows NT 4.0 nachrüstbar
- In Windows 2000 enthalten

© G. Born

ADSI-Beispiel (Benutzer)

```
' Benutzer zu Gruppe hinzufügen
domain = "//Wien"
group = "Benutzer"
user = "Bill"
' Binde an Gruppe (nutze WinNT:)
Set oGroup = GetObject("WinNT:" & Domain & "/" & group)
' Benutzer hinzufügen
oGroup.Add "WinNT:" & Domain & "/" & user

' Benutzerbeschreibung setzen
Set oUser = GetObject("WinNT:" & domain & "/" & user)
oUser.Description = "Neuer Angestellter"
oUser.SetInfo
Set oGroup = Nothing : Set oUser = Nothing
```

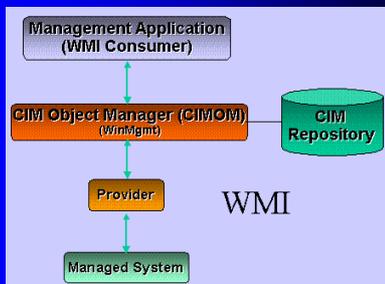
© G. Born

Windows Management Interface (WMI)

- WMI ist
 - Microsofts Web Based Enterprise Management (WBEM)-Implementierung
 - Schnittstelle zum CIM-Repository
 - Erlaubt Rechnerkonfiguration abzufragen und anzupassen

© G. Born

WMI-Hierarchie



© G. Born

WMI-Speicherabfrage

```

' Liefert freien Speicher über WMI ExecQuery-Methode
Set oWMI = _
GetObject("winmgmts:{impersonationLevel=impersonate}!//Rom")
strQuery = "Select FreePhysicalMemory From " & _
           "Win32_OperatingSystem" ' Query für freien Speicher erzeugen
Set oSet = oWMI.ExecQuery(strQuery) ' Query ausführen
For Each oltem in oSet ' Ergebnisse auflisten (max. 1 Wert)
If oltem Is nothing Then
txt = "Freier physikal. Speicher: unbekannt"
Else
txt = "Freier physikal. Speicher: " & _
      FormatNumber(oltem.FreePhysicalMemory, 0) & " Byte"
End if
Next ' Ergebnis zeigen
MsgBox txt, vbOkOnly + vbInformation, "Maschine \\\Rom"
WScript.Quit
  
```

© G. Born

Wie geht's weiter?

- WSH 2.0 Neuerungen
- WSH im Internet
- Literatur

© G. Born

WSH 2.0-Neuerungen

- **WSF-Dateiformat**
 - XML-Format (erlaubt Includes, Referenzen auf Typbibliotheken, mehrere Jobs etc.)
- **WSH 2.0 bringt neue Methoden**
 - z.B. Sleep, AppActivate, SendKeys, LogEvent, Std-IO/Streams

© G. Born

WSF-Dateien (XML)

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<job id="T1" prompt="yes">
<?job error="false" debug="true" logo="false" validate="false" ?>
<script language="VBScript">
<![CDATA[
stop
WScript.Echo "Hallo,,
]]>
</script>
<script language="JScript">
<![CDATA[
debugger;
WScript.Echo ("Hallo1");
]]>
</script>
</job>
  
```

© G. Born

WSH im Internet

Microsofts Scripting Site

msdn.microsoft.com/scripting

WSH Bazaar

www.borncity.de/WSHBazaar (oder)
ourworld.compuserve.com/homepages/Guenter_Born

Newsgroup zum WSH

microsoft.public.scripting.wsh

© G. Born

Literatur

G. Born

Inside Windows Script Host (2.0)

Microsoft Press, München, 2000,

ISBN 3-86063-616-2

© G. Born