Application Engineering Documentation

Windows Script Host: Introduction and Resources

Contents

Microsoft's Windows Script Host

Practical Use of WSH

Because WSH supports VBScript, ASP programmers can immediately apply their programming skills to scripting tasks. Much of the ASP code you have written for an IIS application can be imported into a scripting context with minor modifications. The difference is that rather than launching it from a web browser, it is run by double clicking on a file on your machine, or supplying it as an argument to wscript.exe or script.exe.

WSH scripting is an appropriate solution for system administration tasks, maintenance, writing simple COM objects, and any task which needs to read or write files on an end-user's desktop computer. It will not replace ASP as the framework for serving web pages. At this time we forsee using WSH only for internal programming tasks. WSH requires that the target computer have specialized software installed, something which we cannot expect clients to do any more than we can expect them to have IE rather than Netscape or Web-TV.

The following is a list of tasks that WSH can do on a desktop computer which cannot be done in the ASP/IIS framework. Actually most of the commands can be called in ASP/IIS, but their target would be the server computer (which could cause the commands to fail because of the more stringent security settings).

- Read, write, create, delete, copy files and folders in your filesystem. See the object Scripting.FileSystemObject, and the sample scripts below. When the FileSystem object is used in an ASP/IIS context, it only interacts with the filesystem of the IIS server computer.
- Create content for MS-Excel, MS-Word, Internet Explorer, and open the file in an instance of the application on the local computer. See sample scripts below.
- Read system environment variables on your local computer. See sample scripts below.
- Read from and write to the Windows registry
- Write COM objects. These will not be high-performance solutions; any mission critical solution would be written in C++ or Java at Giant Step. It may be a useful framework for learning COM basics, however.
- Modify the Windows desktop environment in various ways (creating shortcuts, modifying the Start Menu, change screen settings and decorative scheme, etc.). Presumably you would do this when you have too many things to create by hand, for example, to set up a computer with your preferred look-and-feel.
- Network management (mapping drives, setting default printer, etc.). See the Wscript.Network object.
- Interact with user with MsgBox() and InputBox()

In addition, WSH allows you to take advantage of many of the powerful features you are accustomed to in the ASP/IIS environment:

Giant Step Programming Guidelines and Tips

This section will expand as WSH is used more at Giant Step.

For WSH scripts written in VbScript, all the standard Giant Step guidelines for Vbscript programming apply.

Any WSH script that is written for any client-related project (whether for internal or external use) should be submitted to the same testing as any other program. When your program is complete, make a SIR request to have it tested by QA.

When using WSH to run a windows batch command, it may be desirable to save the output. Unfortunately there is no way to redirect the output into a variable in your script. What you need to do is call the command with the output redirected to a file in a known location, and then use the filesystemobject to read this file. See sample in this document.

When a script is purely When a script contains mixed language use (Jscript, Vbscript, XML) the extension of the file must be "**.wsf**". (NOTE: previously the extension was "**.ws**" but this was changed to "**.wsf**" in the final release. Beware that some published books on WSH that were published before the final release still give the extension as "**.ws**".) See sample in this document of mixed language use.

Be aware that many of WSH's capabilities for creating dynamic browser content are Microsoft-centric, i.e. they may require Internet Explorer.

Question: can you edit your WSH file in Visual Interdev to take advantage of the nifty and helpful text coloring scheme? Yes, but only through a sloppy workaround: create an HTML document and surround your WSH code with a <script language=VBSCript></script> tag (or Jscript, as the case may be).

A buggy program can hang or get in an infinite loop. To kill the script, go to the Windows NT Task Manager and kill the process called wscript.exe.

Scripting Engines version 5.1 implements VBScript 5, whereas Giant Step's deveopment and production servers still (as of Jan. 2000) use VBScript 4. Beware of developing version 5 habits in WSH and getting confused when your code doesn't run in ASP.

Note that WSH scripts can run in one of two manners: a Windows mode and a Console mode. When you double click a WSH script, it automatically runs the script with the program WSCRIPT.EXE, which is the Windows mode. This mode uses popup windows for dialog boxes, and allows for other uses of the Windows GUI (for example, launching IE). You can launch the same script from a DOS window by typing *WSCRIPT scriptname*. The Console mode is invoked by calling *CSCRIPT.EXE scriptname*, and causes any output to be written to the console window that called it. Note that if ascript intended to run in Console mode does indeed require interaction, the user must use the Wscript.Echo() facility rather than MsgBox() and InputBox(), because the latter cannot be presented in a DOS Window. Scripts that are run in unattended manner (for example, scheduled tasks like running of DTS scripts) obviously will be called by CSCRIPT, and should not have any interactive features at all (not even Wscript.Echo()).

```
set ts = fso.OpenTextFile(fname, 1, False, 0)
accum = ""
while (not ts.atEndOfStream)
  accum = accum & ts.ReadLine() & VbCrLf
wend
ts.Close()
MsgBox(accum)
```

Run a Windows program with Run()

```
' Here is an excerpt that makes calls to Mtxstop, mtxrereg, and the net
'  facility as though  you entered them from a Command Prompt (DOS
'  window)
'
Option Explicit
Dim WshShell, ans, quit,
Set WshShell = Wscript.CreateObject("Wscript.Shell")
ans = msgbox("Going to run Mtxstop and stop IIS",49,"")
quit = False
if (ans = "1") then
   WshShell.Run ("mtxstop ")
   WshShell.Run ("net stop iisadmin /y ")
   ans = msgbox("Wait until last action completes; compile your object, and hit okay when
done, to restart IIS and run mtxrereg",49,"")
  else
   quit = True
end if
if (not quit and ans = 1) then
   WshShell.Run ("net start w3svc /y ")
   WshShell.Run ("mtxrereg.exe ")
  else
   quit = True
end if
```

Create an MS-Word document

```
' CreateMSwordFile.vbs
'
' Declare the variable.
Option Explicit
Dim objWD , wObj, oFolder
    ' Set the variable (runs new instance of Word.)
    Set objWD = CreateObject("Word.Application")

    ' Add a new document.
    objWD.Documents.Add

    ' Add some text.
    objWD.Selection.TypeText "This is some text."

    ' Save the document on the desktop
    set wObj = CreateObject("WScript.Shell")
    Set oFolder = wObj.SpecialFolders
    objWD.ActiveDocument.SaveAs oFolder.Item("Desktop") & _
```

```
' this is a regular expression matching Admin*.gif
filespec = "Admin.*\.gif"

Set fso = CreateObject("Scripting.FileSystemObject")
Set f = fso.GetFolder(dirName)

' Use the RegExp object (new to VBScript 5.1) to match the right
' filenames
Set re = New RegExp
re.pattern = filespec

If f.Files.Count > 0 then
   MsgBox("Ready to start copying files.  A dialog box will pop " & _
     "up when the copying is done.")
   For Each oldfile In f.Files
      if (re.Test(oldfile.Name)) then
         oldfileFullpath = dirName & "\" & oldfile.Name
         newfile = Replace(oldfile.Name, " ", "")
         newfileFullpath = dirName2 & "\" & newfile
         fso.CopyFile oldfileFullpath, newfileFullpath, True
      end if
   Next
   MsgBox "File copying done"
 else
   MsgBox "No files to copy"
End If
```

Database/ADO calls

```
' UseADO.vbs
'
Option Explicit
Dim rs, objConnection
Set objConnection = CreateObject("ADODB.Connection")
objConnection.Open "EN_Energizer"    ' put your System DSN here
set rs = objConnection.Execute("select count(*) from sweeps_entry")
MsgBox("There are " & rs(0) & " records in EN_Energizer.sweeps_entry")
rs.Close()
set rs = Nothing
objConnection.Close()
set objConnection = Nothing
```

Create an Excel File (flashy demo)

```
' ExcelDemo.vbs
'
' Windows Script Host Sample Script
'
' From the MSDN site,  Copyright (C) 1996-1997 Microsoft Corporation
'
L_Welcome_MsgBox_Message_Text    = "This script demonstrates how to access Excel
using the Windows Scripting Host."
L_Welcome_MsgBox_Title_Text      = "Windows Scripting Host Sample"
Call Welcome()
```

```
    Jglobal = "variable Jglobal declared in JScript";
    function JFoo() {
      return("Function JFoo() declared in JScript");
    }
  </script>
  <script language="VBScript">
    WScript.Echo("VBScript can read " + Jglobal)
    WScript.Echo("VBScript can call " + JFoo())
  </script>
  <script language="JScript">
    WScript.Echo("JScript can read " + Vglobal);
    WScript.Echo("JScript can call " + VFoo())
  </script>
</job>
```

Writing HTML dynamically to a web browser

```
<!--
    DynamicWebContent.wsf

    Note:  this must be saved in a .wsf file -->
<job>
  <script language="VBScript">
    Dim strHtml, ie, n
    n = 0
    strHtml = inputbox("Enter your name:", "Your name","")
    strHtml = "Here is some <b>dynamic</b> web content written " & _
        "by <font color=red size=+5><i>" & strHtml & "</i></font>"
    Set ie = WScript.CreateObject("InternetExplorer.Application","")
    ie.Visible = True
    ie.Navigate "about:blank"
  </script>
  <script language="JScript">
    ie.document.body.innerHTML = strHtml
  </script>
</job>
```

Resources

Windows Script Host

http://msdn.microsoft.com/scripting is the general Microsoft site. See http://msdn.microsoft.com/scripting/default.htm?/scripting/windowshost/docs/reference/default.htm for online documentation of WSH.

Windows Script Host, by Tim Hill, Oct. 1999, Macmillan Technical Publishing; ISBN: 1578701392 *(note: this version is specific to Windows 2000)*

Windows Script Host Programmer's Reference, by Dino Esposito, 1999 WROX Press, ISBN 1-861002-65-3

Special info on controlling cursor location & text color on the screen)

- http://ds.dial.pipex.com/town/gateway/y/yyz81/batch.html

Windows NT Shell Scripting

Published book: Windows NT Shell Scripting by Tim Hill, publ April 1998 by Macmillan Technical Publishing, ISBN 1578700477
Windows NT Resource Kit

Microsoft® Windows® 98 Resource Kit by Microsoft Corporation, publ May, 1998, ISBN = 1-57231-644-6

http://www.winmag.com/people/jruley/ntbat-l.htm:

Resources on Scripting in General, History

http://www.scriptics.com/scripting/scripting.html

http://www.scriptics.com/people/john.ousterhout/scripting.html

Perl with WSH

A system can be quickly set up to run Perl from WSH by purchasing a product from a company called ActiveState. See http://www.activestate.com/