Scripting Tutorial

Writing administrative scripts in VBScript is pretty straightforward. There are basically two things you'll need to learn: VBScript itself and the various COM objects that perform the functions you want to use in your scripts. You can think of the COM objects as the bricks for a building, and think of VBScript as the mortar that holds them together and makes them into a solid structure. Perhaps the best way to learn to use VBScript in your administrative scripts is with plenty of examples. Listing 1 shows a basic script that creates shortcuts on a desktop. This might be useful in a logon script or a script that configures new computers to meet your corporate standards.

Listing 1 Creating Shortcuts

```
'this sample creates two shortcuts on the current user's desktop
'shows how to use the Shell interface from within Script.
'first, we need to create an instance of the shell object
dim objShell
set objShell = WScript.CreateObject("WScript.Shell")

'next, we need to get the path to the special Desktop folder
dim strDesktop
strDesktop = objShell.SpecialFolders("Desktop")

'now, we can create shortcuts on the desktop

'let's do Internet Explorer
dim objShortcut
set objShortcut= objShell.CreateShortcut(strDesktop & "\IE.lnk")

with objShortcut
 .TargetPath = "iexplore.exe "
 .WindowStyle = 1
 .Hotkey = "CTRL+SHIFT+I"
 .Description = "Launch Internet Explorer"
 .WorkingDirectory = strDesktop
 .Save
end with

'let's create a link to my home page
dim objURL
set objURL = objShell.CreateShortcut(strDesktop & "\BrainCore Website.url")
objURL.TargetPath = "http://www.braincore.net"
objURL.Save
```

This script makes heavy use of the Windows Shell COM object, which is a part of Windows's scripting system. This object can provide the actual paths to special folders, such as the `Desktop` folder, and it can create and manipulate shortcuts. The script starts by creating a variable, `objShell`, and assigning it to represent the Windows Shell COM object by using the `Set` command. Next, the script retrieves the path of the `Desktop` folder and saves it in a variable named `strDesktop`. Finally, the script creates two shortcuts, saving their references in variables named `objURL` and `objShortcut`. Once created, the script sets the properties of the shortcuts to point to specific files (Internet Explorer) or URLs (the `brainCore.net` Web site).

Listing 2 is a short example that uses the Active Directory Scripting Interface (ADSI) to connect to a domain controller and modify information in the domain.

Listing 2 Setting Domain Information

```
'first bind to the domain
set objDomain = GetObject("WinNT://MyDomain")
objDomain.Put "MinPasswordLength", 8
objDomain.Put "MinPasswordAge", 10
objDomain.Put "MaxPasswordAge", 45
objDomain.Put "MaxBadPasswordsAllowed", 3
objDomain.Put "PasswordHistoryLength", 8
objDomain.Put "AutoUnlockInterval", 30000
objDomain.Put "LockoutObservationInterval", 30000
objDomain.SetInfo
```

To use this script, be sure to change `MyDomain` to the name of your domain. This script starts by setting an object variable to represent the domain itself and then using the domain's `Put` method to modify properties, such as the minimum password age. Although this script uses ADSI (remember, the AD stands for Active Directory), it will run with a Windows NT domain, too, because it's using ADSI's `WinNT` provider, which is backward-compatible with NT domains.

As you can see, administrative scripting usually involves the creation of a few variables, which you accomplish with the `Dim` statement. You'll often set some of those variables to represent COM objects by using the `Set` statement along with the `CreateObject` command. Finally, you'll use the objects' methods to accomplish tasks, such as the `Put` method or the `CreateShortcut` method.

You can find more documentation about VBScript and these objects online at the MSDN Library: `http://msdn.Microsoft.com/library`. Look under the Platform SDK section.