TechNet Home > Script Center

Scripting: Your First Steps

By The Scripting Guys

A long time ago, even before any of the Scripting Guys were born, people used to walk from one place to another. Not for exercise or recreation, but because they had to get somewhere. Those who were fortunate enough to have other forms of transportation could move along a little more quickly on their horses, camels, elephants, or whatever the local pack animal happened to be. (And to those of you still using any of the preceding as your primary mode of transportation—we'll be thinking of you the next time we're stuck in traffic.)

Then this thing came along known as the horseless carriage. Many people laughed. They said things like "It's just a fad," "Why would I ever want one of those?" and even "That looks too dangerous." Then there were the very few visionaries who said "Wow, cool, I want to try one of those." These last people were the ones who recognized not only how much fun a car could be, but how it could someday save them hours, even days, of travel time in getting from place to place.

People today have experienced some of the same reactions to scripting as those people had to cars all those years ago. "Why would I want to script?"; "Scripting is just for those fanatics who don't want to run their systems like everyone else does"; and even, "Scripts are dangerous." But actually, you don't need to be visionary or adventurous to try scripting, you just need to be the type of person who wants to save some time. (It's still only the fanatics who think it's fun though.) Scripts can not only make your work go faster, they can make your job easier. And once you learn the basic rules of the road, they're not all that difficult to operate. (And there are very few traffic jams.)

On This Page

Step 1

```
C:\>cscript test.vbs
```

This is the very first thing you need to know to start using scripts to manage your Microsoft Windows systems. Well, okay, technically it's the second thing you need to know. The first thing you need to know is how to find the Script Center, but since you've obviously done that we decided to leave out that step.

On the Script Center (and possibly elsewhere on the Internet, but we can't vouch for anyone else), you'll find a lot of pre-made, ready-to-go scripts. A script will look something like this:

```
    Wscript.echo "My very first script."
```

Yes, that one line is a script. It’s a very *simple* script, most are longer than this, but it’s still a script. Now you might be wondering what to do with that one line. It’s very simple:

- Open Notepad.
- Copy the script from your browser and paste it into Notepad.
- Save the script with a .vbs extension, such as test.vbs.

Now just open a command window, navigate to the folder where you saved your file, and type this:

```
    cscript test.vbs
```

If you did this with the script above, you’ll have output that looks like this:

```
    My very first script.
```

> **Note:** There are probably some of you out there saying “Hey, this didn’t work! What’s the deal?” Don’t worry, you’re not alone. A common misstep in following the instructions we just gave you has to do with the Save As functionality. If you select Save As in Notepad and type a filename in the File Name box, Notepad will helpfully append a .txt file extension to whatever name you give it. So if you type in **test.vbs**, you’ll end up with a file named **test.vbs.txt**. There are two easy ways to prevent this, it’s your choice as to which way is easiest for you:
>
> - Surround the filename with double quotes. Instead of typing **test.vbs** in the File Name box, type **“test.vbs”**.
> - Select **All Files** from the Save As Type list box before clicking Save.

Perform these same steps and you can run just about any script you’ll find in the Script Center. Have fun!

Top of page

What is All This Scripting Stuff Anyway?

Oh, you want to know more? Okay, let’s step back a moment (yes, all the way back to step zero) and talk about what scripting is and why you might want to use scripts.

Scripting is just a way to automate getting information to and from your computer (and other computers). Our first script did this: we gave a sentence *to* the computer and got the same sentence back *from* the computer. This may not seem like an especially useful feature, but this was just a first step, and one step doesn’t even get you all the way across the room, let alone out the door. (Okay, one of the Scripting Guys did once live in an apartment where one step could get you across the room. But, for various reasons we won’t go into [something about ice and giant spiders], that apartment wasn’t much more useful than our first script.) There are a number of scenarios where scripts start to get really useful. Here are just a few:

- You have to perform a series of system administration tasks on a regular basis
- You have to perform a series of system administration tasks on several computers
- You want to consolidate and organize the output you get from the computer

- You want to run tasks when you’re not there to interact with the GUI
- You need to ensure the exact same actions are repeated each time a task is run

If you think about some of the work you have to do as a system administrator, you might already be imagining tasks you want to script.

↑ Top of page

But This Stuff Looks Like Code

So maybe you’ve already been looking around the Script Center, possibly even read some of the daily Hey, Scripting Guy! articles, and you’ve seen scripts that look similar to this:

```
strComputer = "."
Set objWMIService = GetObject("winmgmts:\\" & strComputer & "\root\cimv2")
Set colProcesses = objWMIService.ExecQuery _
    ("Select * from Win32_Process Where Name = 'Dfrgntfs.exe'")

If colProcesses.Count = 0 Then
    Wscript.Echo " Dfrgntfs.exe is not running."
Else
    Wscript.Echo " Dfrgntfs.exe is running."
End If
```

You might be thinking, “This scripting stuff looks pretty complicated.” Believe it or not, scripting isn’t just for professional computer programmers. There are much more powerful and complicated tools for programmers to use. Don’t get us wrong, a script can be powerful and complicated, but you can also write some fairly simple scripts that can be very helpful. So don’t let the thought of this stuff being “code” scare you off. There are a lot of things you can do pretty easily with no programming experience at all.

One of the advantages of scripting over other types of programming is that for scripting, everything you need is built into the operating system. We’ll briefly discuss some acronyms such as VBScript, WSH, and WMI a little later, but we’ll tell you right now that all of those things are part of scripting, and they’re all built into Windows. You also don’t need fancy, expensive software to write a script. As you saw already, all you really need is Notepad, which, once again, is built-in.

In addition, scripting is specifically designed to help you administer your operating system. Even the most talented programmer would never attempt to create a full-blown software application, such as Word or Excel, using scripts. Scripts allow you to automate system administration tasks.

Keep in mind that the type of scripting we’re talking about here—and that we deal with almost exclusively on the Script Center—is system administration scripting. There are other types of scripting (such as web scripting), but you’ll have to learn about that somewhere else.

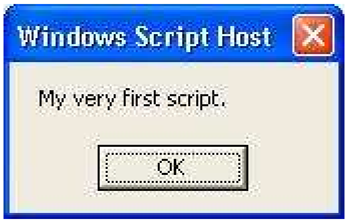
There are other scripting languages besides VBScript, and there are tools you can use to write scripts other than Notepad. But you can investigate all that on your own after you get a little scripting experience. Most of the scripts and examples on the Script Center use VBScript.

IMPORTANT: One last thing about system administration scripting, and it’s a really important thing to know. For most scripts to run, at least scripts that do anything very interesting, you *must* have local administrator rights to the machine the script is running against. Many of the scripts available on the Script Center will fail if you’re not running them as a local administrator.

↑ Top of page

Running Scripts

Enough of the background stuff, let’s get back to running some scripts. We already showed you how to run a script from the command line with Cscript. (If you got excited when you saw “Running Scripts” and came right here, you need to calm down, take a deep breath, and go back to the top and read Step 1.) You can also run scripts just by double-clicking on them from My Computer or Windows Explorer. The difference will be that the output from the script won’t be printed out nicely in the command window, instead it will pop up in a message box. Try this with the script we already created. Just double-click on the test.vbs file. You should see a message box that looks like this:



My very first script.

Well that was easy. Why didn’t we just do that in the first place? What do we need the command window for? Before we answer that, try this: Paste the following script into Notepad, save it with a .vbs extension, and double-click on the file.

```
For i = 1 to 5
    Wscript.echo i
Next
```

What happened? You had to click OK in five different message boxes. Just imagine if your script were returning all the processes running on a computer, or all the computer names in a domain. You could be clicking for quite a while. Running your script with Cscript sends all this output straight to the command window, and you don’t have to deal with all those message boxes.

For more on running scripts, take a look at Running WSH Scripts in the *Windows 2000 Scripting Guide*.

Top of page

What About Those Acronyms?

Because we said earlier we’d mention them, and the Scripting Guys never lie (well, maybe the occasional “Yes, your new haircut looks great”), here’s a very brief definition of some scripting-related technologies, along with the sections of the *Windows 2000 Scripting Guide* that explain them more fully:

- **VBScript – Visual Basic, Scripting Edition**. A scripting language available by default with Microsoft Windows. See VBScript Primer.
- **WSH – Windows Script Host**. The environment in which your scripts run. See WSH Primer.
- **WMI – Windows Management Instrumentation**. A technology that provides you with the resources to manage Windows operating systems through scripting. See WMI Scripting Primer.
- **ADSI – Active Directory Service Interfaces**. A technology that provides you with the resources to manage Active Directory and other directory services through scripting. See ADSI Scripting Primer.

Top of page

Navigating the Script Center

The Script Center provides a number of resources to help you out as you begin scripting and as you become more skilled.

Learn to Script. The best place to start is on the Learn to Script page. Here you’ll find links to the *Windows 2000 Scripting Guide* (a full 1200 page book online), various articles, virtual labs, and on-demand webcasts. In the beginning you’ll probably find viewing the webcasts from Scripting Week 1 to be very helpful.

Script Repository. Another place you’ll probably find yourself spending a lot of time is the Script Center Script

Repository. Here you’ll find thousands of scripts already created for you. Why spend a lot of time working on writing a new script when there might be one out there you can just copy for free? If you don’t see exactly what you’re looking for, you might find something close enough that you can make a simple modification to get what you need. (Keep in mind that the scripts on the Script Center typically don’t do any error checking, so once you get a little more comfortable and your scripts start to get more complex you’ll want to take a look at Error Handling in the *Windows 2000 Scripting Guide*.)

Scripting Tools. The Script Center also has links to numerous tools you can download—for free even—that will help you write scripts.

Scripting For…. If you’re trying to solve a problem in a specific area of system administration, such as Desktop Deployment or Windows Server 2003, take a look at the technology-specific areas of the Script Center.

Fun Zone. And last but certainly not least, if you find yourself with some time on your hands and just want to have some fun (and learn something in the process!), check out Dr. Scripto’s Fun Zone.

⇡ Top of page

You’re On Your Own…Mostly

We want to share one helpful tip with you before we send you out on your own: scripts are *not* case sensitive. There are one or two exceptions that we always point out to you, but for the most part mixing case in a script is purely for readability. So this script (which generates random numbers):

```
intHighNumber = 100
intLowNumber = 1

For i = 1 to 5
    Randomize
    intNumber = Int((intHighNumber - intLowNumber + 1) * Rnd + intLowNumber)
    Wscript.Echo intNumber
Next
```

will run exactly the same as this script:

```
inthighnumber = 100
intlownumber = 1

for i = 1 to 5
    randomize
    intnumber = int((inthighnumber - intlownumber + 1) * rnd + intlownumber)
    wscript.echo intnumber
Next
```

You should now be ready to head into the rest of the Script Center and get to work. If you really get stuck, you can email scripter@microsoft.com and we’ll try to see what we can do to help. Have fun, and good luck!

⇡ Top of page

More Fun for Beginners

We hope you enjoyed this introduction to scripting, and more importantly that it was helpful to you. For more on beginning scripting, check out the **Sesame Script** series!

⇡ Top of page

Manage Your Profile