

# Introduction

Microsoft introduced Visual Basic, Scripting Edition—commonly known as VBScript—in the mid-1990s, positioning it as a native replacement for Windows’ aging command-line batch language, which was based on Microsoft’s earliest operating system, MS-DOS. VBScript was intended to be easy to learn, powerful, and flexible. The language was included as an add-on to Windows 95 and Windows NT 4.0, was an optional installation component included in Windows 98, and was included in all editions of Windows Me, Windows 2000, Windows XP, and Windows Server 2003.

Software developers immediately seized upon VBScript for web programming, particularly in Active Server Pages, Microsoft’s rapid-development programming framework for the web. However, Windows administrators—one of VBScript’s initial target audiences—were left cold. VBScript seemed to be much more complicated than administrators’ beloved MS-DOS-based batch language, and many didn’t see the need to learn an entirely new batch language.

When Windows 2000 and Active Directory came along, however, administrators found that Windows administration had become a great deal more complex. Suddenly, administrators were searching for Resource Kits and other utilities that offered automated administration, especially for repetitive tasks. Active Directory enabled the use of VBScript for logon and logoff scripts, which seemed to promise more advanced-use environment manipulation. At around the same time, Microsoft’s naiveté in releasing a powerful language like VBScript with absolutely no security controls resulted in a huge wave of high-impact VBScript-based viruses, forcing administrators to lock down their environments and remove VBScript as an option both for viruses and for administrative tools.

As a regular speaker at some of the country’s top technical conferences that focus on Windows technologies, including TechMentor, the past few years I’ve given half- and full-day sessions on VBScripting for Windows administrators, and the sessions have been incredibly popular. In these sessions, I try to provide just enough VBScript experience to make scripting possible, and then concentrate on accomplishing common administrative tasks with VBScript. I also cover the security concerns of VBScript and provide administrators with the means for safely using VBScript in their environments. This book is essentially a written form of those sessions, greatly expanded with more coverage of Windows Management Instrumentation and other advanced topics, and with more coverage of VBScript security issues and resolutions.

I’m not out to turn you into a programmer. In fact, one of the real successes of VBScript is that you don’t *need* to be a programmer to use it. Most of what you’ll be doing in this book involves using VBScript to tell Windows to do things for you; you’ll be able to ignore much of VBScript’s complexity, using it as a sort of electronic glue to combine various operating system functions.

It's been four years since the original edition of this book, published as *Managing Windows with VBScript and WMI*. At the time, Windows administrators were really just discovering scripting and its potential to automate administrative tasks; since then, scripting and automation have taken off in a big way. Managers—not just administrators—realize that automation makes better use of skilled technical professionals, freeing them up from boring, repetitive tasks for new projects. That realization has led to the word *scripting* being added to many a high-end job description, and scripting is emerging as one of the most important differentiators between entry-level technicians and experienced professionals.

In the past four years, I've done a tremendous amount of work to promote scripting and education. I produced more than a dozen free Webcasts for Microsoft TechNet (which are still viewable; links can be found on <http://www.ScriptingAnswers.com>), launched a web community for administrative scripting called [www.ScriptingAnswers.com](http://www.ScriptingAnswers.com), created two complete series of training videos for scripting (viewable at <http://www.ScriptingTraining.com>), designed a web search engine specifically for scripting resources (<http://www.SearchScripting.com>), and wrote nearly a half-dozen books on Windows scripting and automation (including two free ones from <http://www.Realtimepublishers.com>). Scripting is here to stay.

## Who Should Read This Book?

The only assumption I have about you is that you already know how to administer some version of Microsoft Windows. You'll find that most of the material in this book is suitable for Windows NT, Windows 2000, Windows Server 2003, and (as it's known as of this writing) Windows "Longhorn" Server environments (that includes the client versions of these operating systems, such as Windows XP and Windows Vista), and it will continue to be useful through future versions of Windows. I do not assume that you have any background in programming, and I'm not going to give you a programming background.

You should have a desire to learn how to use what I call "the batch language of the twenty-first century" and a desire to move away from clumsier—and often more complex—batch files based on the MS-DOS batch language. Although some folks like to refer to batch files as scripts, I don't; and when you see how easy and flexible VBScript is, you'll understand why!

## How to Use This Book

You can read this book in order from the Introduction to the Appendix. However, if you already have some experience with VBScript, or if you just want to dive right in to the more complete sample scripts, you can skip around as much as you want. This book is organized in the same way that I organize my live VBScripting sessions at conferences, so you might feel that it's some time before you really get into the meat of scripting. I assure you, though, that each example in this book—starting in Chapter 1—is focused on Windows administration. You'll get your feet wet right away!

To help you decide where to start, the following sections provide a brief overview of each chapter.

## **Part I: Introduction to Windows Administrative Scripting**

Part I serves as an introduction to the world of scripting and provides you with a methodology for approaching administrative tasks from a scripting standpoint. One of the most difficult parts about producing new scripts from scratch is the “Where do I start?” factor, and this part provides you with a framework for figuring that out every time.

### **Chapter 1: Scripting Concepts and Terminology**

As implied previously, administrative scripting isn’t hard-core programming. Instead, it’s using VBScript as a sort of electronic glue to secure various bits of the Windows operating system together. This chapter introduces you to those various bits and sets the stage with some basic terminology that you’ll use throughout this book.

### **Chapter 2: Running Scripts**

Writing a script isn’t much fun if you can’t run the script! This chapter focuses on the technologies used to execute scripts. You might be surprised to learn how many different Microsoft products support scripting. This chapter shows you how far your scripting skills can really take you and also introduces you to some scripting tools that can make writing and debugging scripts a bit easier.

### **Chapter 3: The Components of a Script**

This chapter presents a complete administrative script and then breaks it down line-by-line to explain its various components. Although this chapter isn’t necessary to learning administrative scripting, it will help you write scripts that are more reliable and easier to troubleshoot.

### **Chapter 4: Designing a Script**

As mentioned previously, one of the toughest aspects about scripting can be figuring out where to start. This chapter provides you with a framework that you can use as a starting point for every new scripting project. This chapter also introduces you to some concepts that many scripting books ignore, such as planning for errors and creating a useful “resource kit” of script components that you can reuse throughout your scripting projects.

## **Part II: VBScript Tutorial**

Part II serves as your official crash course to the VBScript language: just enough to make administration via script a possibility! The best part is that this part doesn’t use the trite “Hello, world” examples that books for software developers often start out with. Instead, every example is useful to you as a Windows administrator. This means you’ll produce simple, useful scripts at the same time you’re learning VBScript. What could be better?

**Chapter 5: Functions, Objects, Variables, and More**

This chapter shows you the basic building blocks of any script and introduces you to some sample scripts that use each building block in a particular administrative task. This is really the meat of administrative scripting, and you'll be able to write useful scripts when you're finished with this chapter.

**Chapter 6: Input and Output**

You can make your scripts more flexible by adding the ability to dynamically change computer, user, and domain names, along with other information. This chapter shows you how your script can collect information it needs to run and dynamically alter itself to take advantage of that information.

**Chapter 7: Manipulating Numbers**

This chapter explains how scripts can manipulate numbers, making it easier to create scripts that work with numeric data, such as user account data. It also introduces you to VBScript's numeric data handling and conversion commands, putting you on the path to some great scripting techniques.

**Chapter 8: Manipulating Strings**

Strings—a fancy word for text data—are at the heart of most scripting tasks. This chapter shows you how VBScript deals with strings and how you can easily integrate them into your scripts.

**Chapter 9: Manipulating Other Types of Data**

Aside from text and numbers, your scripts might need to deal with dates, times, bytes, and other forms of data to accomplish specific administrative tasks. This chapter shows you how VBScript handles these other data types and how you can use them in your own scripts.

**Chapter 10: Controlling the Flow of Execution**

The best administrative scripts can respond to changing conditions with internal logic, called control-of-flow. This chapter shows you how your scripts can be made to evaluate various conditions and respond accordingly, perform repetitive tasks, and much more.

**Chapter 11: Built-in Scripting Objects**

Much of VBScript's power comes from its capability to join various operating system objects, and this chapter introduces you to your first set of those objects. You'll learn how to manipulate network information, map drives, and much more—pretty much everything you need to write effective logon scripts.

**Chapter 12: Working with the File System**

A common use of scripting is to manipulate files and folders, and this chapter introduces you to the VBScript `FileSystemObject`, which provides a complete object model for working with the file system. You'll learn to build a utility that scans Internet Information Services (IIS) log files for error messages, a useful script for any environment!

**Chapter 13: Putting It All Together: Creating Your First Script from Scratch**

This is where you put everything from Part II together. You'll create a script that rotates IIS log files, keeping the past 30 days worth of files in a special archive folder. This chapter guides you through the complete process of designing, writing, testing, and troubleshooting the script. In fact, it deliberately introduces some logic errors into the script so that you can see the debugging process in action.

**Part III: Windows Management Instrumentation and Active Directory Services Interface**

With the glue of VBScript under your belt, this part dives into the two most powerful technologies for administering Windows: Windows Management Instrumentation (WMI) and the Active Directory Services Interface (ADSI). These technologies provide administrative access to, and control over, nearly every aspect of the Windows operating system, from Windows NT to Windows Server 2003.

**Chapter 14: Working with ADSI Providers**

Despite its name, ADSI isn't just for Active Directory. This chapter shows you how ADSI can be used to interface with NT, Active Directory, Novell NDS, Exchange Server, and other types of directory services. This chapter provides some basic examples of the types of tasks you can perform with ADSI to get you started.

**Chapter 15: Manipulating Domains**

With the ADSI basics out of the way, this chapter focuses on manipulating domain information in a script. You'll learn how to query domain information, modify domain policies such as password length, and much more.

**Chapter 16: Manipulating Users and Groups**

This chapter shows you how to write scripts that query and modify user and group information. This is one of the most common tasks you'll perform with VBScript, and this chapter includes plenty of useful examples.

**Chapter 17: Understanding WMI**

WMI provides a hook into just about every portion of the Windows operating system, making it an incredibly useful tool for administrative scripts. This chapter introduces you to WMI and shows you a preview of what you can use it for in your environment.

**Chapter 18: Querying Basic WMI Information**

Do you want to find out which users in your organization have a Pentium 4 computer? This chapter shows you how to write your own basic WMI queries, including those that involve remote machines. You'll also learn basic WMI manipulation, which lets you modify local and remote machine settings from within a script.

**Chapter 19: Querying Complex WMI Information**

Some WMI queries are more complex, such as querying the IP addresses from multiple network adapters in multiple remote computers. This chapter provides clear examples of these more complex WMI tasks, helping you learn to write enterprise management scripts.

**Chapter 20: Putting It All Together: Your First WMI/ADSI Script**

This is where it all comes together. This chapter walks you through the process of designing, writing, testing, and debugging a complete WMI/ADSI script from scratch. You'll finish this chapter with a concrete example of the administrative capabilities of these technologies, and then you'll be ready to start writing your own scripts.

**Chapter 21: Testing and Debugging WMI and ADSI Queries**

Getting the perfect WMI or ADSI query is critical to the success of your scripts, so this chapter focuses on tools you can use to develop those queries more interactively, test your queries, and have them fully refined before pasting them into your scripts.

**Part IV: Advanced Scripting Techniques**

As you become a more experienced scripter, you'll be ready to start saving time and be more secure, with advanced techniques like script encryption, scripting components, script security, and so forth. This part of the book gives you a comprehensive look at each of these technologies and shows you how to put them into use in your own environment.

**Chapter 22: Modular Script Programming**

If you find yourself cutting and pasting code—or worse, retyping it—this is the chapter for you. This chapter introduces you to modular scripting concepts, which make it easier to reuse code between various scripts, saving you time and effort! By way of example, this chapter starts with a complex script that contains lots of useful code and then breaks it down into easily reused modules.

**Chapter 23: Scripts Packaging and Protection**

Are you worried that others will peek into your scripts and steal your ideas? Script packaging and other techniques help protect your scripts from both Peeping Toms and potential misuse, so this chapter shows you how to set up, deploy, and use script packages within your environment.

**Chapter 24: Scripting Security**

Some folks think Microsoft made a huge mistake when it included VBScript in the Windows operating system, but others disagree. Properly configured, scripting can be as safe as any other type of application. This chapter explains scripting security concepts and introduces you to the tools that can make scripting a safe and valuable part of any computing environment.

**Chapter 25: Introduction to HTML Applications**

HTML Applications, or HTAs, provide a way to mix VBScript and Hypertext Markup Language (HTML) code to produce graphical scripts that look almost like full Windows applications. They're a great way to produce tools that you plan to share with less-experienced users or administrators. This chapter gives you a quick start in building HTAs, along with an explanation of how they differ from more traditional VBScript projects.

**Chapter 26: Debugging Tips, Tools, and Techniques**

By now, you'll have seen your fair share of script bugs, and so this chapter shows you how to prevent them from happening, find them quickly when they do happen, and squash them just as quickly so that you can get on with your scripting.

**Part V: Ready-to-Run Examples**

This part is a great way to wrap up the book—with a whole section on ready-made sample scripts that you can start using in your own environment. In addition, these scripts—like every other script in this book—have complete, line-by-line explanations, making them a perfect reference guide as you start to create your own scripts from scratch.

**Chapter 27: Logon and Logoff Scripts**

This chapter presents more complex logon and logoff scripts and gives you some ideas for how scripting can make these important scripts more effective. Of course, the line-by-line explanations make each script a useful reference for customizing your own scripts.

**Chapter 28: Windows and Domain Administration Scripts**

Automating domain administration is probably one of the big reasons you started looking at scripting in the first place, so this chapter presents a number of examples of tasks that scripts can perform. The detailed explanations with each script will help you rip them apart and customize them for your own use.

**Chapter 29: Network Administration Scripts**

Network administration is ideally suited for scripting, and this chapter provides a handful of examples that show you what's possible. The line-by-line explanations make it easy to put these into use in your own environment.

**Chapter 30: WMI and ADSI Scripts**

These can be the toughest scripts to write because of the complexity and flexibility of WMI and ADSI. This chapter provides you with several ready-to-use scripts for common tasks, such as querying WMI, creating users and groups, and more. These scripts can be easily modified and incorporated into your own scripts, saving you scripting time!

## Appendix

### Appendix: Administrator's Quick Script Reference

One of the toughest parts about VBScript is that it contains so much functionality. It's usually pretty easy to figure out what you want a script to do; the tough part is often figuring out how to make VBScript do it! This appendix provides an alphabetical list of common tasks and gives the VBScript commands that perform each task. You can use this reference along with the VBScript documentation to make designing and writing scripts much easier.

## Preparing to Use This Book

Before you dive in, you should make sure that your computers are ready for VBScript. Fortunately, any computer with Windows 2000 or later is ready to go out of the box, and this book assumes that you're doing your development work on either a Windows 2000-, Windows XP-, or Windows Server 2003-based computer.

## Typographical Elements

Books on programming can benefit a great deal from easy-to-understand typestyles and elements like the ones explained here. These typestyles and elements are designed to make the text easier to follow and to call your attention to special concerns.

Monospaced type will be used to set off material that should be typed into the computer. For example, "select Run from the Start menu, type `wbemtest`, and click OK" sets off the menu selection, and what you need to type onscreen.

Blocks of code and code lines that appear within the text appear in a monospaced font, as in, "To change the contents of a variable, you can use `Var1 = Trim(Var1)`."

### TIP

Tips provide shortcuts and other "insider advice" about scripting that you'll find valuable.

---

### NOTE

Notes provide cautions and other clarifications that will help you avoid problems or further clarify complex concepts.

---



You'll also be directed to material that more thoroughly explains particular concepts, VBScript commands, and so forth. Although you might not be a big fan of flipping back and forth through a book, these cross-references allow you to remain focused within each chapter and guide you to more detailed explanations, when appropriate.

Finally, there are times when it is necessary to present an extended explanation of something that isn't critical to the task at hand. In those cases, a sidebar is included. A sidebar is a cue that the information is useful, but it's not really key to the main text; you're welcome to skip the sidebar and come back to it later.

### Sidebars

Sidebars make it easier to cover slightly off-topic information without distracting you from the main text.

## Sample Scripts

Obviously, a book on scripting is going to have many code listings. To make these as useful as possible, each sample script is presented in a listing by itself with no comments.

### LISTING P1 A Sample Script

```
'Get the user's name
sName = InputBox("What is your name?")

'Display the user's name
MsgBox "Your name is " & sName
```

After each script, any changes you might need to make to get the script running in your environment, such as changing computer or domain names, are presented. You'll find each complete script included on the CD that accompanies this book. Each chapter has a separate folder and the script files are named with their listing number for easy reference.

## Sample Scripts—Explained

For each script in this book, a line-by-line explanation of the script is included, so that you understand exactly what's going on. For example:

First, the sample script displays a dialog box where the user can type his name. By default, this dialog box includes an OK and Cancel button; this script does not provide any way to detect the Cancel button, so it is assumed the user will type something and click OK.

```
'Get the user's name
sName = InputBox("What is your name?")
```

Finally, the script uses the `MsgBox` statement to redisplay the user's name. Notice the use of the ampersand operator (`&`) to tack on the contents of the variable `sName`, which stores whatever the user typed into the input box.

```
'Display the user's name  
MsgBox "Your name is " & sName
```

Walk-throughs like this one will help you become more familiar with VBScript, what each command does, and exactly how each sample script works.