Introduction

We shall not cease from exploration
And the end of all our exploring
Will be to arrive where we started
And know the place for the first time.

—T. S. Eliot

My goal in writing *Microsoft Windows Vista Unleashed* is to cover the good, the bad, and, yes, even the ugly of Windows Vista. In particular, I give you complete coverage of the intermediate-to-advanced features of Windows Vista. This means that I bypass basic topics, such as wielding the mouse, in favor of more complex operations, such as working with the Registry, maintaining and troubleshooting your system, networking, and getting around the Internet.

I’ve tried to keep the chapters focused on the topic at hand and unburdened with long-winded theoretical discussions. However, there are plenty of situations in which you won’t be able to unleash the full power of Windows Vista and truly understand what’s going on unless you have a solid base on which to stand. In these cases, I’ll give you whatever theory and background you need to get up to speed. From there, I’ll get right down to brass tacks without any further fuss and bother.

Who Should Read This Book

All writers write with an audience in mind. Actually, I’m not sure whether that’s true for novelists and poets and the like, but it *should* be true for any technical writer who wants to create a useful and comprehensible book. Here are the members of my own imagined audience:

- **IT professionals**—These brave souls must decide whether to move to Vista, work out deployment issues, and support the new Vista desktops. The whole book has information related to your job and Vista.

- **Power users**—These elite users get their power via knowledge. With that in mind, this book extends the Windows power user’s know-how by presenting an exhaustive account of everything that’s new and improved in Windows Vista.

- **Business users**—If your company is thinking of or has already committed to moving to Vista, you need to know what you, your colleagues, and your staff are getting into. You also want to know what Vista will do to improve your productivity and make your life at the office easier. You learn all of this and more in this book.
Road warriors—if you travel for a living, you probably want to know what Vista brings to the remote computing table. Will you be able to synchronize data, connect to the network, and manage power better than before? What other new notebook features can be found in Vista? You’ll find out in this book.

Small business owners—if you run a small or home business, you probably want to know whether Vista will give you a good return on investment. Will it make it easier to set up and maintain a network? Will Vista computers be more stable? Will your employees be able to collaborate easier? The answer turns out to be “Yes” for all of these questions, and I’ll show you why.

Multimedia users—if you use your computer to listen to music or radio stations, watch TV, work with digital photographs, edit digital movies, or burn CDs and DVDs, you’ll be interested to know that Vista has a handful of new features that affect all of these activities.

Also, to keep the chapters uncluttered, I’ve made a few assumptions about what you know and what you don’t know:

- I assume that you have knowledge of rudimentary computer concepts such as files and folders.
- I assume that you’re familiar with the basic Windows skills: mouse maneuvering, dialog box negotiation, pull-down menu jockeying, and so on.
- I assume that you can operate peripherals attached to your computer, such as the keyboard and printer.
- I assume that you’ve used Windows for a while and are comfortable with concepts such as toolbars, scrollbars, and, of course, windows.
- I assume that you have a brain that you’re willing to use and a good supply of innate curiosity.

How This Book Is Organized

To help you find the information you need, this book is divided into seven parts that group related tasks. The next few sections offer a summary of each part.

Part I: Unleashing Day-to-Day Windows Vista

Part I takes your basic, workaday Windows chores and reveals their inner mysteries, allowing you to become more productive. After an initial chapter on what’s new in Vista, topics include the myriad ways to get Windows Vista off the ground (Chapter 2), how to use Windows Vista to work with files and folders (Chapter 3), getting the most out of file types (Chapter 4), installing and running applications (Chapter 5), working with user accounts (Chapter 6), dealing with digital media (Chapter 7), using Contacts, Calendar, and faxing (Chapter 8), and Vista’s mobile computing tools (Chapter 9).
Part II: Unleashing Essential Windows Vista Power Tools
The chapters in Part II get your advanced Windows Vista education off to a flying start by covering the ins and outs of four important Vista power tools: Control Panel and group policies (Chapter 10), the Registry (Chapter 11), and the Windows Script Host (Chapter 12).

Part III: Unleashing Windows Vista Customization and Optimization
In Part III, you dive into the deep end of advanced Windows work: customizing the interface (Chapter 13), performance tuning (Chapter 14), maintaining Windows Vista (Chapter 15), troubleshooting problems (Chapter 16), and working with devices (Chapter 17).

Part IV: Unleashing Windows Vista for the Internet
Part IV shows you how to work with Windows Vista’s Internet features. You learn how to get the most out of a number of Internet services, including the Web (Chapter 18), email (Chapter 19), and newsgroups (Chapter 20). I close this part with an extensive look at the Internet security and privacy feature that come with Windows Vista (Chapter 21).

Part V: Unleashing Windows Vista Networking
To close out the main part of this book, Part V takes an in-depth look at Windows Vista’s networking features. You learn how to set up a small network (Chapter 22), how to access and use that network (Chapter 23), and how to access your network from remote locations (Chapter 24).

Part VI: Appendixes
To further your Windows Vista education, Part VI presents a few appendixes that contain extra goodies. You’ll find complete list of Windows Vista shortcut keys (Appendix A), a detailed look at using the Windows Vista command prompt (Appendix B), and a batch file primer (Appendix C).

What’s New in the Second Edition
Microsoft Windows Vista Unleashed 2E includes coverage of the new features that are part of Windows Vista Service Pack 1 (SP1). Most of those features are under-the-hood tweaks that improve Vista’s performance, reliability, security, application compatibility, and driver support. I’ll talk about those where appropriate, and of course I’ll also talk about any changes that SP1 makes to the Vista interface. (For example, in Chapter 15, “Maintaining Your Windows Vista System,” I talk about the new Disk Defragmenter feature that enables you to select which disks get defragmented.)
## Conventions Used in This Book

To make your life easier, this book includes various features and conventions that help you get the most out of this book and Windows Vista itself:

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Things you type</td>
<td>Whenever I suggest that you type something, what you type appears in a <strong>bold monospace</strong> font.</td>
</tr>
<tr>
<td>Filenames, folder names, and code</td>
<td>These things appear in a monospace font.</td>
</tr>
<tr>
<td>Commands</td>
<td>Commands and their syntax use the monospace font as well. Command placeholders (which stand for what you actually type) appear in an <em>italic monospace</em> font.</td>
</tr>
<tr>
<td>Pull-down menu commands</td>
<td>I use the following style for all application menu commands: <em>Menu, Command</em>, where <em>Menu</em> is the name of the menu that you pull down and <em>Command</em> is the name of the command you select. Here’s an example: File, Open. This means that you pull down the File menu and select the Open command.</td>
</tr>
<tr>
<td>Code continuation character</td>
<td>When a line of code is too long to fit on only one line of this book, it is broken at a convenient place and continued to the next line. The continuation of the line is preceded by a code continuation character (➥). You should type a line of code that has this character as one long line without breaking it.</td>
</tr>
</tbody>
</table>

This book also uses the following boxes to draw your attention to important (or merely interesting) information:

**NOTE**

The Note box presents asides that give you more information about the current topic. These tidbits provide extra insights that give you a better understanding of the task. In many cases, they refer you to other sections of the book for more information.
TIP
The Tip box tells you about Windows Vista methods that are easier, faster, or more efficient than the standard methods.

CAUTION
The all-important Caution box tells you about potential accidents waiting to happen. There are always ways to mess things up when you’re working with computers. These boxes help you avoid at least some of the pitfalls.
CHAPTER 12

Programming the Windows Script Host

In Appendix C, “Automating Windows XP with Batch Files,” you learn how to tame the command prompt by creating batch files—small, executable text files that run one or more commands. You’ll see that with a little ingenuity and a dash of guile, it’s possible to make batch files perform some interesting and useful tasks. Indeed, for many years, batch files were the only way to automate certain kinds of tasks. Unfortunately, the batch file world is relentlessly command-line–oriented. So, with the exception of being able to launch Windows programs, batch files remain ignorant of the larger Windows universe.

If you’re looking to automate a wider variety of tasks in Windows, you need to supplement your batch file knowledge with scripts that can deal with the Registry, shortcuts, files, and network drives, and that can even interact with Windows programs via Automation. The secret to these powerful scripts is the Windows Script Host (WSH). This chapter introduces you to the Windows Script Host, shows you how to execute scripts, and runs through the various elements in the Windows Script Host object model.

WSH: Your Host for Today’s Script

As you might know, Internet Explorer is really just an empty container application that’s designed to host different data formats, including ActiveX controls, various file formats (such as Microsoft Word documents and Microsoft Excel worksheets), and several ActiveX scripting engines. A scripting engine is a dynamic link library (DLL) that provides programmatic support for a particular scripting language. Internet Explorer supports two such scripting engines: VBScript (VBScript.dll) and JavaScript.
This enables web programmers to write small programs—scripts—that interact with the user, control the browser, set cookies, open and close windows, and more. Although these scripting engines don’t offer full-blown programmability (you can’t compile scripts, for example), they do offer modern programming structures such as loops, conditionals, variables, objects, and more. In other words, they’re a huge leap beyond what a mere batch file can do.

The Windows Script Host is also a container application, albeit a scaled-down application in that its only purpose in life is to host scripting engines. Right out of the box, the Windows Script Host supports both the VBScript and JavaScript engines. However, Microsoft designed the Windows Script Host to be a universal host that can support any ActiveX-based scripting engine. Therefore, there are also third-party vendors offering scripting engines for languages such as Perl, Tcl, and Rexx.

The key difference between Internet Explorer’s script hosting and the Windows Script Host is the environment in which the scripts run. Internet Explorer scripts are web page–based, so they control and interact with either the web page or the web browser. The Windows Script Host runs scripts within the Windows Vista shell or from the command prompt, so you use these scripts to control various aspects of Windows. Here’s a sampling of the things you can do:

- Execute Windows programs
- Create and modify shortcuts
- Use Automation to connect and interact with Automation-enabled applications such as Microsoft Word, Outlook, and Internet Explorer
- Read, add, and delete Registry keys and items
- Access the VBScript and JavaScript object models, which give access to the file system, runtime error messages, and more
- Use pop-up dialog boxes to display information to the user, and determine which button the user clicked to dismiss the dialog box
- Read environment variables, which are system values that Vista keeps in memory, such as the folder into which Vista is installed—the %SystemRoot% environment variable—and the name of the computer—the %ComputerName% environment variable
- Deal with network resources, including mapping and unmapping network drives, accessing user data (such as the username and user domain), and connecting and disconnecting network printers

Clearly, we’ve gone way beyond batch files!

What about speed? After all, you wouldn’t want to load something that’s the size of Internet Explorer each time you need to run a simple script. That’s not a problem because, as I’ve said, the Windows Script Host does nothing but host scripting engines, so
it has much less memory overhead than Internet Explorer. That means that your scripts run quickly. For power users looking for a Windows-based batch language, the Windows Script Host is a welcome tool.

NOTE
This chapter does not teach you how to program in either VBScript or JavaScript and, in fact, assumes that you’re already proficient in one or both of these languages. If you’re looking for a programming tutorial, my VBA for the 2007 Microsoft Office System (Que, 2007) is a good place to start (VBScript is a subset of VBA—Visual Basic for Applications). For JavaScript, try my Special Edition Using JavaScript (Que, 2001).

Scripts and Script Execution

Scripts are simple text files that you create using Notepad or some other text editor. You can use a word processor such as WordPad to create scripts, but you must make sure that you save these files using the program’s Text Only document type. For VBScript, a good alternative to Notepad is the editor that comes with either Visual Basic or any program that supports VBA (such as the Office suite). Just remember that VBScript is a subset of VBA (which is, in turn, a subset of Visual Basic), so it does not support all objects and features.

In a web page, you use the <script> tag to specify the scripting language you’re using, as in this example:

```
<SCRIPT LANGUAGE="VBScript">
```

With the Windows Script Host, the script file’s extension specifies the scripting language:

- For VBScript, save your text files using the .vbs extension (which is registered as the following file type: VBScript Script File).
- For JavaScript, use the .js extension (which is registered as the following file type: JScript Script File).

As described in the next three sections, you have three ways to run your scripts: by launching the script files directly, by using WScript.exe, or by using CScript.exe.

Running Script Files Directly

The easiest way to run a script from within Windows is to launch the .vbs or .js file directly. That is, you either double-click the file in Windows Explorer or type the file’s path and name in the Run dialog box. Note, however, that this technique does not work at the command prompt. For that, you need to use the CScript program described a bit later.
Using WScript for Windows-Based Scripts

The .vbs and .js file types have an open method that’s associated with WScript (WScript.exe), which is the Windows-based front-end for the Windows Script Host. In other words, launching a script file named MyScript.vbs is equivalent to entering the following command in the Run dialog box:

```
wscript myscript.vbs
```

The WScript host also defines several parameters that you can use to control how the script executes. Here’s the full syntax:

```
  //S //T:ss //X
```

- **filename** Specifies the filename, including the path of the script file, if necessary.
- **arguments** Specifies optional arguments required by the script. An argument is a data value that the script uses as part of its procedures or calculations.
- **//B** Runs the script in batch mode, which means script errors and Echo method output lines are suppressed. (I discuss the Echo method later in this chapter.)
- **//D** Enables Active Debugging. If an error occurs, the script is loaded into the Microsoft Script Debugger (if it’s installed) and the offending statement is highlighted.
- **//E:engine** Executes the script using the specified scripting engine, which is the scripting language to use when running the script.
- **//H:host** Specifies the default scripting host. For host, use either CScript or WScript.
- **//I** Runs the script in interactive mode, which displays script errors and Echo method output lines.
- **//Job:xxxx** In a script file that contains multiple jobs, executes only the job with id attribute equal to xxxx.
- **//S** Saves the specified WScript arguments as the default for the current user; uses the following Registry key to save the settings:
  HKCU\Software\Microsoft\Windows Script Host\Settings
- **//T:ss** Specifies the maximum time in seconds (ss) that the script can run before it shuts down automatically.
- **//X** Executes the entire script in the Microsoft Script Debugger (if it’s installed).
For example, the following command runs MyScript.vbs in batch mode with a 60-second maximum execution time:

wscript myscript.vbs //B //TT:60

---

### Creating Script Jobs

A script **job** is a section of code that performs a specific task or set of tasks. Most script files contain a single job. However, it’s possible to create a script file with multiple jobs. To do this, first surround the code for each job with the `<script>` and `</script>` tags, and then surround those with the `<job>` and `</job>` tags. In the `<job>` tag, include the `id` attribute and set it to a unique value that identifies the job. Finally, surround all the jobs with the `<package>` and `</package>` tags. Here’s an example:

```xml
<package>
  <job id="A">
    <script language="VBScript">
      WScript.Echo "This is Job A."
    </script>
  </job>

  <job id="B">
    <script language="VBScript">
      WScript.Echo "This is Job B."
    </script>
  </job>

</package>
```

Save the file using the `.wsf` (Windows Script File) extension.

---

**NOTE**

If you write a lot of scripts, the Microsoft Script Debugger is an excellent programming tool. If there’s a problem with a script, the debugger can help you pinpoint its location. For example, the debugger enables you to step through the script’s execution one statement at a time. If you don’t have the Microsoft Script Debugger, you can download a copy from msdn.microsoft.com/scripting.

---

### Using CScript for Command-Line Scripts

The Windows Script Host has a second host front-end application called CScript (`CScript.exe`), which enables you to run scripts from the command line. In its simplest form, you launch CScript and use the name of the script file (and its path, if required) as a parameter, as in this example:

`cscript myscript.vbs`
The Windows Script Host displays the following banner and then executes the script:

Microsoft (R) Windows Script Host Version 5.7 for Windows
Copyright Microsoft Corporation. All rights reserved.

As with WScript, the CScript host has an extensive set of parameters you can specify:

➥ //S //T:ss //X //U //LOGO //NOLOGO

This syntax is almost identical to that of WScript, but adds the following three parameters:

//LOGO Displays the Windows Script Host banner at startup
//NOLOGO Hides the Windows Script Host banner at startup
//U Uses Unicode for redirected input/output from the console

**Script Properties and .wsh Files**

In the last two sections, you saw that the WScript and CScript hosts have a number of parameters you can specify when you execute a script. It’s also possible to set some of these options by using the properties associated with each script file. To see these properties, right-click a script file and then click Properties. In the properties sheet that appears, display the Script tab, shown in Figure 12.1. You have two options:

- **Stop Script After Specified Number of Seconds**—If you activate this check box, Windows shuts down the script after it has run for the number of seconds specified in the associated spin box. This is useful for scripts that might hang during execution. For example, a script that attempts to enumerate all the mapped network drives at startup might hang if the network is unavailable.

- **Display Logo When Script Executed in Command Console**—As you saw in the previous section, the CScript host displays some banner text when you run a script at the command prompt. If you deactivate this check box, the Windows Script Host suppresses this banner (unless you use the //LOGO parameter).

When you make changes to these properties, the Windows Script Host saves your settings in a new file that has the same name as the script file, except with the .wsh (Windows Script Host Settings) extension. For example, if the script file is MyScript.vbs, the settings are stored in MyScript.wsh. These .wsh files are text files organized into sections, much like .ini files. Here’s an example:

```
[ScriptFile]
Path=C:\Users\Paul\Documents\Scripts\Popup1.vbs
[Options]
Timeout=0
DisplayLogo=1
```
FIGURE 12.1  In a script file’s properties sheet, use the Script tab to set some default options for the script.

To use these settings when running the script, use either WScript or CScript and specify the name of the .wsh file:

wscript myscript.wsh

NOTE
Rather than setting properties for individual scripts, you might prefer to set global properties that apply to the WScript host itself. Those global settings then apply to every script that runs using the WScript host. To do this, run WScript.exe without any parameters. This displays the properties sheet for WScript, which contains only the Script tab shown in Figure 12.1. The settings you choose in the properties sheet are stored in the following Registry key:

HKLM\Software\Microsoft\Windows Script Host\Settings

Programming Objects

Although this chapter isn’t a programming primer per se, I’d like to take some time now to run through a few quick notes about programming objects. This will serve you well throughout the rest of the chapter as I take you on a tour of the Windows Script Host object model.
The dictionary definition of an object is “anything perceptible by one or more of the senses, especially something that can be seen and felt.” In scripting, an object is an application element that exposes an interface to the programmer, who can then perform the programming equivalent of seeing and feeling:

- You can make changes to the object’s properties (this is the seeing part).
- You can make the object perform a task by activating a method associated with the object (this is the feeling part).

**Working with Object Properties**

Every programmable object has a defining set of characteristics. These characteristics are the object’s properties, and they control the appearance and position of the object. For example, the `WScript` object (the top-level Windows Script Host object) has an Interactive property that determines whether the script runs in interactive mode or batch mode.

When you refer to a property, you use the following syntax:

```
Object.Property
```

- `Object` is the name of the object
- `Property` is the name of the property with which you want to work

For example, the following expression refers to the Interactive property of the `WScript` object:

```
WScript.Interactive
```

**Setting the Value of a Property**

To set a property to a certain value, you use the following syntax:

```
Object.Property = value
```

Here, `value` is an expression that specifies the value to which you want to set the property. As such, it can be any of the scripting language’s recognized data types, which usually include the following:

- A numeric value
- A string value, enclosed in double quotation marks (such as "My Script Application")
- A logical value (in VBScript: True or False; in JavaScript: true or false)
For example, the following VBScript statement tells the Windows Script Host to run the script using interactive mode:

```vbnet
WScript.Interactive = True
```

**Returning the Value of a Property**

Sometimes you need to know the current setting of a property before changing the property or performing some other action. You can find out the current value of a property by using the following syntax:

```vbnet
variable = Object.Property
```

Here, `variable` is a variable name or another property. For example, the following statement stores the current script mode (batch or interactive) in a variable named `currentMode`:

```vbnet
currentMode = WScript.Interactive
```

**Working with Object Methods**

An object’s properties describe what the object is, whereas its **methods** describe what the object **does**. For example, the `WScript` object has a `Quit` method that enables you to stop the execution of a script.

How you refer to a method depends on whether the method requires any arguments. If it doesn’t, the syntax is similar to that of properties:

```vbnet
Object.Method
```

- **Object**  The name of the object
- **Method**  The name of the method you want to run

For example, the following statement shuts down a script:

```vbnet
WScript.Quit
```

If the method requires arguments, you use the following syntax:

```vbnet
Object.Method(Argument1, Argument2, ...)
```

**NOTE**

In VBScript, the parentheses around the argument list are necessary only if you’ll be storing the result of the method in a variable or object property. In JavaScript, the parentheses are always required.
For example, the `WshShell` object has a `RegWrite` method that you use to write a key or value to the Registry. (I discuss this object and method in detail later in this chapter; see “Working with Registry Entries.”) Here’s the syntax:

```javascript
WshShell.RegWrite strName, anyValue[, strType]
```

- `strName` The name of the Registry key or value
- `anyValue` The value to write, if `strName` is a Registry value
- `strType` The data type of the value

### Argument Naming Conventions

When presenting method arguments in this chapter, I’ll follow Microsoft’s naming conventions, including the use of the following prefixes for the argument names:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>any</td>
<td>Any type</td>
</tr>
<tr>
<td>b</td>
<td>Boolean</td>
</tr>
<tr>
<td>int</td>
<td>Integer</td>
</tr>
<tr>
<td>nat</td>
<td>Natural numbers</td>
</tr>
<tr>
<td>obj</td>
<td>Object</td>
</tr>
<tr>
<td>str</td>
<td>String</td>
</tr>
</tbody>
</table>

For many object methods, not all the arguments are required. In the `RegWrite` method, for example, the `strName` and `anyValue` arguments are required, but the `strType` argument is not. Throughout this chapter, I differentiate between required and optional arguments by surrounding the optional arguments with square brackets—for example, `[strType]`.

For example, the following statement creates a new value named `Test` and sets it equal to `Foo`:

```javascript
WshShell.RegWrite "HKCU\Software\Microsoft\Windows Script Host\Test",
➥ "Foo", "REG_SZ"
```

### Assigning an Object to a Variable

If you’re using JavaScript, you assign an object to a variable using a standard variable assignment:

```javascript
var variableName = ObjectName
```

- `variableName` The name of the variable
- `ObjectName` The object you want to assign to the variable
In VBScript, you assign an object to a variable by using the `Set` statement. `Set` has the following syntax:

```
Set variableName = ObjectName
```

<table>
<thead>
<tr>
<th>variableName</th>
<th>The name of the variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectName</td>
<td>The object you want to assign to the variable</td>
</tr>
</tbody>
</table>

You’ll see later on that you must often use Automation to access external objects. For example, if you want to work with files and folders in your script, you must access the scripting engine object named `FileSystemObject`. To get this access, you use the `CreateObject` method and store the resulting object in a variable, like so:

```
Set fs = CreateObject("Scripting.FileSystemObject")
```

### Working with Object Collections

A **collection** is a set of similar objects. For example, `WScript.Arguments` is the set of all the arguments specified on the script’s command line. Collections are objects, too, so they have their own properties and methods, and you can use these properties and methods to manipulate one or more objects in the collection.

The members of a collection are **elements**. You can refer to individual elements by using an **index**. For example, the following statement refers to the first command-line argument (collection indexes always begin at 0):

```
WScript.Arguments(0)
```

If you don’t specify an element, the Windows Script Host assumes that you want to work with the entire collection.

### VBScript: Using For Each...Next Loops for Collections

As you might know, VBScript provides the `For...Next` loop that enables you to cycle through a chunk of code a specified number of times. For example, the following code loops 10 times:

```
For counter = 1 To 10
    Code entered here is repeated 10 times
Next counter
```

A useful variation on this theme is the `For Each...Next` loop, which operates on a collection of objects. You don’t need a loop counter because VBScript loops through the individual elements in the collection and performs on each element whatever operations are inside the loop. Here’s the structure of the basic `For Each...Next` loop:

```
For Each element In collection
    [statements]
Next
```
The following code loops through all the arguments specified on the script’s command line and displays each one:

```javascript
For Each arg In WScript.Arguments
    WScript.Echo arg
Next
```

### JavaScript: Using Enumerators and `for` Loops for Collections

To iterate through a collection in JavaScript, you must do two things: create a new Enumerator object and use a `for` loop to cycle through the enumerated collection.

To create a new Enumerator object, use the `new` keyword to set up an object variable (where `collection` is the name of the collection you want to work with):

```javascript
var enum = new Enumerator(collection)
```

Then set up a special `for` loop:

```javascript
for (; !enumerator.atEnd(); enumerator.moveNext())
{
    [statements];
}
```

The Enumerator object’s `moveNext` method runs through the elements in the collection, whereas the `atEnd` method shuts down the loop after the last item has been processed. The following code loops through all the arguments specified on the script’s command line and displays each one:

```javascript
var args = new Enumerator(WScript.Arguments);
for (; !args.atEnd(); args.moveNext())
{
    WScript.Echo(args.item());
}
```

### Programming the WScript Object

The `WScript` object represents the Windows Script Host applications (`WScript.exe` and `CScript.exe`). You use this object to get and set certain properties of the scripting host, as well as to access two other objects: `WshArguments` (the `WScript` object’s `Arguments` property)
Displaying Text to the User

The `WScript` object method that you'll use most often is the `Echo` method, which displays text to the user. Here’s the syntax:

```wscript
WScript.Echo [Argument1, Argument2,...]
```

Here, `Argument1`, `Argument2`, and so on, are any number of text or numeric values that represent the information you want to display to the user. In the Windows-based host (`WScript.exe`), the information displays in a dialog box; in the command-line host (`CScript.exe`), the information displays at the command prompt (much like the `command-line ECHO` utility).

Shutting Down a Script

You use the `WScript` object’s `Quit` method to shut down the script. You can also use `Quit` to have your script return an error code by using the following syntax:

```wscript
WScript.Quit [intErrorCode]
```

`intErrorCode` An integer value that represents the error code you want to return

You could then call the script from a batch file and use the `ERRORLEVEL` environment variable to deal with the return code in some way. (See Appendix C for more information on `ERRORLEVEL`.)

Scripting and Automation

Applications such as Internet Explorer and Word come with (or `expose`, in the jargon) a set of objects that define various aspects of the program. For example, Internet Explorer has an `Application` object that represents the program as a whole. Similarly, Word has a `Document` object that represents a Word document. By using the properties and methods that come with these objects, it’s possible to programmatically query and manipulate the applications. With Internet Explorer, for example, you can use the `Application` object’s `Navigate` method to send the browser to a specified web page. With Word, you can read a `Document` object’s `Saved` property to see whether the document has unsaved changes.

This is powerful stuff, but how do you get at the objects that these applications expose? You do that by using a technology called `Automation`. Applications that support Automation implement object libraries that expose the application’s native objects to Automation-aware programming languages. Such applications are `Automation servers`, and the applications that manipulate the server’s objects are `Automation controllers`. The Windows Script Host is an Automation controller that enables you to write script code to control any server’s objects.
This means that you can use an application’s exposed objects more or less as you use the Windows Script Host objects. With just a minimum of preparation, your script code can refer to and work with the Internet Explorer Application object, or the Microsoft Word Document object, or any of the hundreds of other objects exposed by the applications on your system. (Note, however, that not all applications expose objects. Windows Mail and most of the built-in Windows Vista programs—such as WordPad and Paint—do not expose objects.)

Creating an Automation Object with the CreateObject Method

The WScript object’s CreateObject method creates an Automation object (specifically, what programmers call an instance of the object). Here’s the syntax:

```
WScript.CreateObject(strProgID)
```

- `strProgID` A string that specifies the Automation server application and the type of object to create. This string is a programmatic identifier, which is a label that uniquely specifies an application and one of its objects. The programmatic identifier always takes the following form:

  `AppName.ObjectType`

  Here, `AppName` is the Automation name of the application and `ObjectType` is the object class type (as defined in the Registry’s HKEY_CLASSES_ROOT key). For example, here’s the programmatic ID for Word:

  `Word.Application`

Note that you normally use `CreateObject` within a Set statement, and that the function serves to create a new instance of the specified Automation object. For example, you could use the following statement to create a new instance of Word’s Application object:

```
Set objWord = CreateObject("Word.Application")
```

You need to do nothing else to use the Automation object. With your variable declared and an instance of the object created, you can use that object’s properties and methods directly. Listing 12.1 shows a VBScript example (you must have Word installed for this to work).
LISTING 12.1 A VBScript Example That Creates and Manipulates a Word Application Object

```vbnet
' Create the Word Application object
Set objWord = WScript.CreateObject("Word.Application")
' Create a new document
objWord.Documents.Add
' Add some text
' Save the document
objWord.ActiveDocument.Save
' We're done, so quit Word
objWord.Quit
```

This script creates and saves a new Word document by working with Word’s Application object via Automation. The script begins by using the CreateObject method to create a new Word Application object, and the object is stored in the objWord variable. From there, you can wield the objWord variable just as though it were the Word Application object.

For example, the objWord.Documents.Add statement uses the Documents collection’s Add method to create a new Word document, and the InsertBefore method adds some text to the document. The Save method then displays the Save As dialog box so that you can save the new file. With the Word-related chores complete, the Application object’s Quit method runs to shut down Word. For comparison, Listing 12.2 shows a JavaScript procedure that performs the same tasks.
LISTING 12.2  A JavaScript Example That Creates and Manipulates a Word Application Object

```javascript
// Create the Word Application object
var objWord = WScript.CreateObject("Word.Application");

// Create a new document
objWord.Documents.Add();

// Add some text

// Save the document
objWord.ActiveDocument.Save();

// We're done, so quit Word
objWord.Quit();
```

Making The Automation Server Visible
The `CreateObject` method loads the object, but doesn’t display the Automation server unless user interaction is required. For example, you see Word’s Save As dialog box when you run the `Save` method on a new document (as in Listings 12.1 and 12.2). Not seeing the Automation server is the desired behavior in most Automation situations. However, if you do want to see what the Automation server is up to, set the `Application` object’s `Visible` property to `True`, as in this example:

```javascript
objWord.Visible = True
```

Working with an Existing Object Using the `GetObject` Method
If you know that the object you want to work with already exists or is already open, the `CreateObject` method isn’t the best choice. In the example in the previous section, if Word is already running, the code will start a second copy of Word, which is a waste of resources. For these situations, it’s better to work directly with the existing object. To do that, use the `GetObject` method:

```javascript
WScript.GetObject(strPathname, [strProgID])
```

`strPathname`  The pathname (drive, folder, and filename) of the file you want to work with (or the file that contains the object you want to work with). If you omit this argument, you have to specify the `strProgID` argument.
strProgID The programmatic identifier that specifies the Automation server application and the type of object to work with (that is, the AppName.ObjectType class syntax).

Listing 12.3 shows a VBScript procedure that puts the GetObject method to work.

LISTING 12.3 A VBScript Example That Uses the GetObject Method to Work with an Existing Instance of a Word Document Object

```vbnet
' Get the Word Document object

' Get the word count
WScript.Echo objDoc.Name & " has " & objDoc.Words.Count & " words."

' We're done, so quit Word
objDoc.Application.Quit
```

The GetObject method assigns the Word Document object named GetObject.doc to the objDoc variable. After you’ve set up this reference, you can use the object’s properties and methods directly. For example, the Echo method uses objDoc.Name to return the filename and objDoc.Words.Count to determine the number of words in the document.

Note that although you’re working with a Document object, you still have access to Word’s Application object. That’s because most objects have an Application property that refers to the Application object. In the script in Listing 12.3, for example, the following statement uses the Application property to quit Word:

objDoc.Application.Quit

Exposing VBScript and JavaScript Objects

One of the most powerful uses for scripted Automation is accessing the object models exposed by the VBScript and JavaScript engines. These models expose a number of objects, including the local file system. This enables you to create scripts that work with files, folders, and disk drives, read and write text files, and more. You use the following syntax to refer to these objects:

Scripting.ObjectType

Scripting is the Automation name of the scripting engine, and ObjectType is the class type of the object.
Programming the FileSystemObject

FileSystemObject is the top-level file system object. For all your file system scripts, you begin by creating a new instance of FileSystemObject:

In VBScript:

```vbscript
define_set fs = WScript.CreateObject("Scripting.FileSystemObject")
```

In JavaScript:

```javascript
var fs = WScript.CreateObject("Scripting.FileSystemObject");
```

Here’s a summary of the file system objects you can access via Automation and the top-level FileSystemObject:

- **Drive**—This object enables you to access the properties of a specified disk drive or UNC network path. To reference a Drive object, use either the Drives collection (discussed next) or the FileSystemObject object’s GetDrive method. For example, the following VBScript statement references drive C:

  ```vbscript
define_set objFS = WScript.CreateObject("Scripting.FileSystemObject")
define_set objDrive = objFS.GetDrive("C:\")
```

- **Drives**—This object is the collection of all available drives. To reference this collection, use the FileSystemObject object’s Drives property:

  ```vbscript
define_set objFS = WScript.CreateObject("Scripting.FileSystemObject")
define_set objDrives = objFS.Drives
```

- **Folder**—This object enables you to access the properties of a specified folder. To reference a Folder object, use either the Folders collection (discussed next) or the FileSystemObject object’s GetFolder method:

  ```vbscript
define_set objFS = WScript.CreateObject("Scripting.FileSystemObject")
define_set objFolder = objFS.GetFolder("C:\My Documents")
```

- **Folders**—This object is the collection of subfolders within a specified folder. To reference this collection, use the Folder object’s Subfolders property:

  ```vbscript
define_set objFS = WScript.CreateObject("Scripting.FileSystemObject")
define_set objFolder = objFS.GetFolder("C:\Windows")
define_set objSubfolders = objFolder.Subfolders
```
File—This object enables you to access the properties of a specified file. To reference a File object, use either the Files collection (discussed next) or the FileSystemObject object’s GetFile method:

```vbscript
Set objFS = WScript.CreateObject("Scripting.FileSystemObject")
Set objFile = objFS.GetFile("c:\autoexec.bat")
```

- Files—This object is the collection of files within a specified folder. To reference this collection, use the Folder object’s Files property:

```vbscript
Set objFS = WScript.CreateObject("Scripting.FileSystemObject")
Set objFolder = objFS.GetFolder("C:\Windows")
Set objFiles = objFolder.Files
```

- TextStream—This object enables you to use sequential access to work with a text file. To open a text file, use the FileSystemObject object’s OpenTextFile method:

```vbscript
Set objFS = WScript.CreateObject("Scripting.FileSystemObject")
Set objTS = objFS.OpenTextFile("C:\Boot.ini")
```

Alternatively, you can create a new text file by using the FileSystemObject object’s CreateTextFile method:

```vbscript
Set objFS = WScript.CreateObject("Scripting.FileSystemObject")
Set objTS = objFS.CreateTextFile("C:\Boot.ini")
```

Either way, you end up with a TextStream object, which has various methods for reading data from the file and writing data to the file. For example, the following script reads and displays the text from C:\Boot.ini:

```vbscript
Set objFS = WScript.CreateObject("Scripting.FileSystemObject")
Set objTS = objFS.OpenTextFile("C:\Boot.ini")
strContents = objTS.ReadAll
WScript.Echo strContents
objTS.Close
```

**Programming the WshShell Object**

WshShell is a generic name for a powerful object that enables you to query and interact with various aspects of the Windows shell. You can display information to the user, run applications, create shortcuts, work with the Registry, and control Windows’ environment variables. The next few sections discuss each of those useful tasks.
Referencing the WshShell Object

WshShell refers to the Shell object exposed via the Automation interface of WScript. Therefore, you must use CreateObject to return this object:

```vbscript
Set objWshShell = WScript.CreateObject("WScript.Shell")
```

From here, you can use the objWshShell variable to access the object’s properties and methods.

Displaying Information to the User

You saw earlier that the WScript object’s Echo method is useful for displaying simple text messages to the user. You can gain more control over the displayed message by using the WshShell object’s Popup method. This method is similar to the MsgBox function used in Visual Basic and VBA in that it enables you to control both the dialog box title and the buttons displayed, as well as to determine which of those buttons the user pressed. Here’s the syntax:

```vbscript
WshShell.Popup(strText, [nSecondsToWait], [strTitle], [intType])
```

- **WshShell**  
The WshShell object.
- **strText**  
The message you want to display in the dialog box. You can enter a string up to 1,024 characters long.
- **nSecondsToWait**  
The maximum number of seconds the dialog box will be displayed.
- **strTitle**  
The text that appears in the dialog box title bar. If you omit this value, Windows Script Host appears in the title bar.
- **intType**  
A number or constant that specifies, among other things, the command buttons that appear in the dialog box (see the next section). The default value is 0.

For example, the following statements display the dialog box shown in Figure 12.2:

```vbscript
Set objWshShell = WScript.CreateObject("WScript.Shell")
objWshShell.Popup "Couldn't find Memo.doc!", , "Warning"
```

**TIP**

For long messages, VBScript wraps the text inside the dialog box. If you prefer to create your own line breaks, use VBScript’s Chr function and the carriage return character (ASCII 13) between each line:

```vbscript
WshShell.Popup "First line" & Chr(13) & "Second line"
```

For JavaScript, use \n instead:

```javascript
WshShell.Popup("First line\nSecond line");
```
Setting the Style of the Message

The default Popup dialog box displays only an OK button. You can include other buttons and icons in the dialog box by using different values for the \texttt{intType} parameter. Table 12.1 lists the available options.

<table>
<thead>
<tr>
<th>VBScript Constant</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{Buttons}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\texttt{vbOKOnly}</td>
<td>0</td>
<td>Displays only an OK button. This is the default.</td>
</tr>
<tr>
<td>\texttt{vbOKCancel}</td>
<td>1</td>
<td>Displays the OK and Cancel buttons.</td>
</tr>
<tr>
<td>\texttt{vbAbortRetryIgnore}</td>
<td>2</td>
<td>Displays the Abort, Retry, and Ignore buttons.</td>
</tr>
<tr>
<td>\texttt{vbYesNoCancel}</td>
<td>3</td>
<td>Displays the Yes, No, and Cancel buttons.</td>
</tr>
<tr>
<td>\texttt{vbYesNo}</td>
<td>4</td>
<td>Displays the Yes and No buttons.</td>
</tr>
<tr>
<td>\texttt{vbRetryCancel}</td>
<td>5</td>
<td>Displays the Retry and Cancel buttons.</td>
</tr>
<tr>
<td>\texttt{Icons}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\texttt{vbCritical}</td>
<td>16</td>
<td>Displays the Critical Message icon.</td>
</tr>
<tr>
<td>\texttt{vbQuestion}</td>
<td>32</td>
<td>Displays the Warning Query icon.</td>
</tr>
<tr>
<td>\texttt{vbExclamation}</td>
<td>48</td>
<td>Displays the Warning Message icon.</td>
</tr>
<tr>
<td>\texttt{vbInformation}</td>
<td>64</td>
<td>Displays the Information Message icon.</td>
</tr>
<tr>
<td>\texttt{Default Buttons}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\texttt{vbDefaultButton1}</td>
<td>0</td>
<td>The first button is the default (that is, the button selected when the user presses Enter).</td>
</tr>
<tr>
<td>\texttt{vbDefaultButton2}</td>
<td>256</td>
<td>The second button is the default.</td>
</tr>
<tr>
<td>\texttt{vbDefaultButton3}</td>
<td>512</td>
<td>The third button is the default.</td>
</tr>
</tbody>
</table>

You derive the \texttt{intType} argument in one of two ways:

- By adding the values for each option
- By using the VBScript constants separated by plus signs (+)

The script in Listing 12.4 shows an example and Figure 12.3 shows the resulting dialog box.
LISTING 12.4  A VBScript Example That Uses the Popup Method to Display the Dialog Box Shown in Figure 12.3

```vbscript
' First, set up the message
strText = "Are you sure you want to copy" & Chr(13)
strText = strText & "the selected files to drive A?"
strTitle = "Copy Files"
intType = vbYesNoCancel + vbQuestion + vbDefaultButton2
'
' Now display it
Set objWshShell = WScript.CreateObject("WScript.Shell")
intResult = objWshShell.Popup(strText, ,strTitle, intType)
```

FIGURE 12.3 The dialog box that’s displayed when you run the script.

Here, three variables—strText, strTitle, and intType—store the values for the Popup method’s strText, strTitle, and intType arguments, respectively. In particular, the following statement derives the intType argument:

```vbscript
intType = vbYesNoCancel + vbQuestion + vbDefaultButton2
```

You also could derive the intType argument by adding up the values that these constants represent (3, 32, and 256, respectively), but the script becomes less readable that way.

### Getting Return Values from the Message Dialog Box

A dialog box that displays only an OK button is straightforward. The user either clicks OK or presses Enter to remove the dialog from the screen. The multibutton styles are a little different, however; the user has a choice of buttons to select, and your script should have a way to find out which button the user chose, which enables it to decide what to do next, based on the user’s selection. You do this by storing the Popup method’s return value in a variable. Table 12.2 lists the seven possible return values.
TABLE 12.2 The Popup Method’s Return Values

<table>
<thead>
<tr>
<th>VBScript Constant</th>
<th>Value</th>
<th>Button Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>vbOK</td>
<td>1</td>
<td>OK</td>
</tr>
<tr>
<td>vbCancel</td>
<td>2</td>
<td>Cancel</td>
</tr>
<tr>
<td>vbAbort</td>
<td>3</td>
<td>Abort</td>
</tr>
<tr>
<td>vbRetry</td>
<td>4</td>
<td>Retry</td>
</tr>
<tr>
<td>vbIgnore</td>
<td>5</td>
<td>Ignore</td>
</tr>
<tr>
<td>vbYes</td>
<td>6</td>
<td>Yes</td>
</tr>
<tr>
<td>vbNo</td>
<td>7</td>
<td>No</td>
</tr>
</tbody>
</table>

To process the return value, you can use an If...Then...Else or Select Case structure to test for the appropriate values. For example, the script shown earlier used a variable called intResult to store the return value of the Popup method. Listing 12.5 shows a revised version of the script that uses a VBScript Select Case statement to test for the three possible return values.

LISTING 12.5 A Script That Uses a Select Case Statement to Process the Popup Method’s Return Value

```vbs
' First, set up the message
strText = "Are you sure you want to copy" & Chr(13)
strText = strText & "the selected files to drive A?"
strTitle = "Copy Files"
intType = vbYesNoCancel + vbQuestion + vbDefaultButton2

' Now display it
Set objWshShell = WScript.CreateObject("WScript.Shell")
intResult = objWshShell.Popup(strText, ,strTitle, intType)

' Process the result
Select Case intResult
    Case vbYes
        WScript.Echo "You clicked ""Yes""!"
    Case vbNo
        WScript.Echo "You clicked ""No""!"
    Case vbCancel
        WScript.Echo "You clicked ""Cancel""!"
End Select
```
Running Applications

When you need your script to launch another application, use the Run method:

\[
\text{WshShell.Run strCommand, [intWindowStyle], [bWaitOnReturn]}
\]

**WshShell**

The WshShell object.

**strCommand**

The name of the file that starts the application. Unless the file is in the Windows folder, you should include the drive and folder to make sure that the script can find the file.

**intWindowStyle**

A constant or number that specifies how the application window will appear:

<table>
<thead>
<tr>
<th>intWindowStyle</th>
<th>Window Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Hidden</td>
</tr>
<tr>
<td>1</td>
<td>Normal size with focus</td>
</tr>
<tr>
<td>2</td>
<td>Minimized with focus (this is the default)</td>
</tr>
<tr>
<td>3</td>
<td>Maximized with focus</td>
</tr>
<tr>
<td>4</td>
<td>Normal without focus</td>
</tr>
<tr>
<td>6</td>
<td>Minimized without focus</td>
</tr>
</tbody>
</table>

**bWaitOnReturn**

A logical value that determines whether the application runs asynchronously. If this value is True, the script halts execution until the user exits the launched application; if this value is False, the script continues running after it has launched the application.

Here’s an example:

```vbscript
Set objWshShell = WScript.CreateObject("WScript.Shell")
objWshShell.Run "Control.exe Inetcpl.cpl", 1, True
```

This Run method launches Control Panel’s Internet Properties dialog box.

**NOTE**

To learn more about launching individual Control Panel icons using Control.exe, refer to “Operating Control Panel” in Chapter 10, “Using Control Panel and Group Policies.”

Working with Shortcuts

The Windows Script Host enables your scripts to create and modify shortcut files. When writing scripts for other users, you might want to take advantage of this capability to display shortcuts for new network shares, Internet sites, instruction files, and so on.
Creating a Shortcut

To create a shortcut, use the CreateShortcut method:

```vbs
WshShell.CreateShortcut(strPathname)
```

- **WshShell** - The WshShell object.
- **strPathname** - The full path and filename of the shortcut file you want to create.

Use the .lnk extension for a file system (program, document, folder, and so on) shortcut; use the .url extension for an Internet shortcut.

The following example creates and saves a shortcut on a user's desktop:

```vbs
Set WshShell = objWScript.CreateObject("WScript.Shell")
Set objShortcut = objWshShell.CreateShortcut("C:\Users\Paul\Desktop\test.lnk")
objShortcut.Save
```

Programming the **WshShortcut** Object

The CreateShortcut method returns a WshShortcut object. You can use this object to manipulate various properties and methods associated with shortcut files.

This object contains the following properties:

- **Arguments** - Returns or sets a string that specifies the arguments used when launching the shortcut. For example, suppose that the shortcut’s target is the following:

  ```vbs
  C:\Windows\Notepad.exe C:\Boot.ini
  ```

  In other words, this shortcut launches Notepad and loads the Boot.ini file. In this case, the Arguments property would return the following string:

  ```vbs
  C:\Boot.ini
  ```

- **Description** - Returns or sets a string description of the shortcut.

- **FullName** - Returns the full path and filename of the shortcut’s target. This will be the same as the strPathname value used in the CreateShortcut method.

- **Hotkey** - Returns or sets the hotkey associated with the shortcut. To set this value, use the following syntax:

  ```vbs
  WshShortcut.Hotkey = strHotKey
  ```

  - **WshShortcut** - The WshShortcut object.
  - **strHotKey** - A string value of the form Modifier+Keyname, where Modifier is any combination of Alt, Ctrl, and Shift, and Keyname is one of A through Z or 0 through 12.
For example, the following statement sets the hotkey to Ctrl+Alt+7:

```vbscript
objShortcut.Hotkey = "Ctrl+Alt+7"
```

- **IconLocation**—Returns or sets the icon used to display the shortcut. To set this value, use the following syntax:

  ```vbscript
  WshShortcut.IconLocation = strIconLocation
  ```

  **WshShortcut**
  - The WshShortcut object.
  **strIconLocation**
  - A string value of the form `Path,Index`, where
    - `Path` is the full pathname of the icon file and
    - `Index` is the position of the icon within the file (where the first icon is 0).

  Here’s an example:

  ```vbscript
  objShortcut.IconLocation = "C:\Windows\System32\Shell32.dll,21"
  ```

- **TargetPath**—Returns or sets the path of the shortcut’s target.

- **WindowStyle**—Returns or sets the window style used by the shortcut’s target. Use the same values outlined earlier for the `Run` method’s `intWindowStyle` argument.

- **WorkingDirectory**—Returns or sets the path of the shortcut’s working directory.

**NOTE**

If you’re working with Internet shortcuts, bear in mind that they support only two properties: `FullName` and `TargetPath` (the URL target).

The `WshShortcut` object also supports two methods:

- **Save**—Saves the shortcut file to disk.

- **Resolve**—Uses the shortcut’s `TargetPath` property to look up the target file. Here’s the syntax:

  ```vbscript
  WshShortcut.Resolve = intFlag
  ```

  **WshShortcut**
  - The WshShortcut object.
  **intFlag**
  - Determines what happens if the target file is not found:

<table>
<thead>
<tr>
<th>intFlag</th>
<th>What Happens</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Nothing</td>
</tr>
<tr>
<td>2</td>
<td>Windows continues to search subfolders for the target file</td>
</tr>
<tr>
<td>4</td>
<td>Updates the <code>TargetPath</code> property if the target file is found in a new location</td>
</tr>
</tbody>
</table>
Listing 12.6 shows a complete example of a script that creates a shortcut.

Listing 12.6  A Script That Creates a Shortcut File

```vbs
Set objWshShell = WScript.CreateObject("WScript.Shell")
Set objShortcut = objWshShell.CreateShortcut("C:\Users\Paul\\Desktop\Edit BOOT.INI.lnk")
With objShortcut
  .TargetPath = "C:\Windows\Notepad.exe"
  .Arguments = "C:\Boot.ini"
  .WorkingDirectory = "C:\"
  .Description = "Opens BOOT.INI in Notepad"
  .Hotkey = "Ctrl+Alt+7"
  .IconLocation = "C:\Windows\System32\Shell32.dll,21"
  .WindowStyle = 3
  .Save
End With
```

Working with Registry Entries

You’ve seen throughout this book that the Registry is one the most crucial data structures in Windows. However, the Registry isn’t a tool that only Windows yields. Most 32-bit applications make use of the Registry as a place to store setup options, customization values the user selected, and much more. Interestingly, your scripts can get in on the act as well. Not only can your scripts read the current value of any Registry setting, but they can also use the Registry as a storage area. This enables you to keep track of user settings, recently used files, and any other configuration data that you’d like to save between sessions. This section shows you how to use the WshShell object to manipulate the Registry from within your scripts.

Reading Settings from the Registry

To read any value from the Registry, use the WshShell object’s RegRead method:

```
WshShell.RegRead(strName)
```

<table>
<thead>
<tr>
<th>WshShell</th>
<th>The WshShell object.</th>
</tr>
</thead>
<tbody>
<tr>
<td>strName</td>
<td>The name of the Registry value or key that you want to read. If strName ends with a backslash (), RegRead returns the default value for the key; otherwise, RegRead returns the data stored in the value. Note, too, that strName must begin with one of the following root key names:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Short Name</th>
<th>Long Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKCR</td>
<td>HKEY_CLASSES_ROOT</td>
</tr>
<tr>
<td>HKCU</td>
<td>HKEY_CURRENT_USER</td>
</tr>
<tr>
<td>Short Name</td>
<td>Long Name</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>HKLM</td>
<td>HKEY_LOCAL_MACHINE</td>
</tr>
<tr>
<td>N/A</td>
<td>HKEY_USERS</td>
</tr>
<tr>
<td>N/A</td>
<td>HKEY_CURRENT_CONFIG</td>
</tr>
</tbody>
</table>

The script in Listing 12.7 displays the name of the registered owner of this copy of Windows XP.

**LISTING 12.7 A Script That Reads the RegisteredOwner Setting from the Registry**

```vbscript
Set objWshShell = WScript.CreateObject("WScript.Shell")
strSetting = "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\RegisteredOwner"
strRegisteredUser = objWshShell.RegRead(strSetting)
WScript.Echo strRegisteredUser
```

**Storing Settings in the Registry**

To store a setting in the Registry, use the WshShell object’s RegWrite method:

```
WshShell.RegWrite strName, anyValue [, strType]
```

- **WshShell**: The WshShell object.
- **strName**: The name of the Registry value or key that you want to set. If `strName` ends with a backslash (\), RegWrite sets the default value for the key; otherwise, RegWrite sets the data for the value. `strName` must begin with one of the root key names detailed in the RegRead method.
- **anyValue**: The value to be stored.
- **strType**: The data type of the value, which must be one of the following: REG_SZ (the default), REG_EXPAND_SZ, REG_DWORD, or REG_BINARY.

The following statements create a new key named ScriptSettings in the HKEY_CURRENT_USER root:

```vbscript
Set objWshShell = WScript.CreateObject("WScript.Shell")
objWshShell.RegWrite "HKCU\ScriptSettings\", ""
```

The following statements create a new value named NumberOfReboots in the HKEY_CURRENT_USER\ScriptSettings key, and set this value to 1:

```vbscript
Set objWshShell = WScript.CreateObject("WScript.Shell")
objWshShell.RegWrite "HKCU\ScriptSettings\NumberOfReboots", 1, "REG_DWORD"
```
Deleting Settings from the Registry
If you no longer need to track a particular key or value setting, use the `RegDelete` method to remove the setting from the Registry:

```
WshShell.RegDelete(strName)
```

- `WshShell` The WshShell object.
- `strName` The name of the Registry value or key that you want to delete. If `strName` ends with a backslash (`\`), `RegDelete` deletes the key; otherwise, `RegDelete` deletes the value. `strName` must begin with one of the root key names detailed in the `RegRead` method.

To delete the `NumberOfReboots` value used in the previous example, you would use the following statements:

```vbs
Set objWshShell = WScript.CreateObject("WScript.Shell")
objWshShell.RegDelete "HKCU\ScriptSettings\NumberOfReboots"
```

Working with Environment Variables
Windows Vista keeps track of a number of environment variables that hold data such as the location of the Windows folder, the location of the temporary files folder, the command path, the primary drive, and much more. Why would you need such data? One example would be for accessing files or folders within the main Windows folder. Rather than guessing that this folder is C:\Windows, it would be much easier to just query the `%SystemRoot%` environment variable. Similarly, if you have a script that accesses files in a user's My Documents folder, hard-coding the username in the file path is inconvenient because it means creating custom scripts for every possible user. Instead, it would be much easier to create just a single script that references the `%UserProfile%` environment variable. This section shows you how to read environment variable data within your scripts.

The defined environment variables are stored in the `Environment` collection, which is a property of the `WshShell` object. Windows Vista environment variables are stored in the "Process" environment, so you reference this collection as follows:

```
WshShell.Environment("Process")
```

Listing 12.8 shows a script that runs through this collection, adds each variable to a string, and then displays the string.
LISTING 12.8  A Script That Displays the System’s Environment Variables

Set objWshShell = WScript.CreateObject("WScript.Shell")
' Run through the environment variables
'
strVariables = ""
For Each objEnvVar In objWshShell.Environment("Process")
    strVariables = strVariables & objEnvVar & Chr(13)
Next
WScript.Echo strVariables

Figure 12.4 shows the dialog box that appears (your mileage may vary).

![Windows Script Host](image)

**FIGURE 12.4** A complete inventory of a system’s environment variables.

If you want to use the value of a particular environment variable, use the following syntax:

```wscript
WshShell.Environment("Process").("strName")
```

`WshShell`  The `WshShell` object

`strName`  The name of the environment variable
Listing 12.9 shows a revised version of the script from Listing 12.6 to create a shortcut. In this version, the **Environment** collection is used to return the value of the %UserProfile% variable, which is used to contrast the path to the current user’s Desktop folder.

**LISTING 12.9  A Script That Creates a Shortcut File Using an Environment Variable**

```vbscript
Set objWshShell = WScript.CreateObject("WScript.Shell")
strUserProfile = objWshShell.Environment("Process")("UserProfile")
Set objShortcut = objWshShell.CreateShortcut(strUserProfile & "\Desktop\Edit BOOT.INI.lnk")
With objShortcut
  .TargetPath = "C:\Windows\Notepad.exe "
  .Arguments = "C:\Boot.ini"
  .WorkingDirectory = "C:\"
  .Description = "Opens BOOT.INI in Notepad"
  .Hotkey = "Ctrl+Alt+7"
  .IconLocation = "C:\Windows\System32\Shell32.dll,21"
  .WindowStyle = 3
  .Save
End With
```

**Programming the WshNetwork Object**

WshNetwork is a generic name for an object that enables you to work with various aspects of the Windows network environment. You can determine the computer name and username, you can enumerate the mapped network drives, you can map new network drives, and more. The next couple of sections show you how to work with this object.

**Referencing the WshNetwork Object**

WshNetwork refers to the Network object exposed via the Automation interface of WScript. This means you use CreateObject to return this object, as shown here:

```vbscript
Set objWshNetwork = WScript.CreateObject("WScript.Network")
```

From here, you use the WshNetwork variable to access the object’s properties and methods.

**WshNetwork Object Properties**

The WshNetwork object supports three properties:

- **ComputerName**
  Returns the network name of the computer

- **UserDomain**
  Returns the network domain name of the current user

- **UserName**
  Returns the username of the current user
Mapping Network Printers

The WshNetwork object supports several methods for working with remote printers. For example, to map a network printer to a local printer resource, use the WshNetwork object’s AddWindowsPrinterConnection method:

```
WshNetwork.AddWindowsPrinterConnection strPrinterPath;
```

Here’s an example:

```
Set objWshNetwork = WScript.CreateObject("WScript.Network")
objWshNetwork.AddWindowsPrinterConnection \"\ZEUS\printer\";
```

To remove a remote printer mapping, use the WshNetwork object’s RemovePrinterConnection method:

```
WshNetwork.RemovePrinterConnection strPrinterPath [, bForce] [, bUpdateProfile]
```

Here’s an example:

```
Set objWshNetwork = WScript.CreateObject("WScript.Network")
objWshNetwork.RemovePrinterConnection \"\ZEUS\inkjet\"
```

Mapping Network Drives

The WshNetwork object supports several methods for mapping network drives. To map a shared network folder to a local drive letter, use the WshNetwork object’s MapNetworkDrive method:

```
WshNetwork.MapNetworkDrive strLocalName, strRemoteName, bUpdateProfile, strUser, strPassword
```

Here’s an example:

```
Set objWshNetwork = WScript.CreateObject("WScript.Network")
objWshNetwork.RemovePrinterConnection \"\ZEUS\inkjet\"
```
If True, the drive mapping is stored in the user’s profile.

Use this value to enter a username that might be required to map the remote share (if you’re logged on as a user who doesn’t have the proper permissions, for example).

Use this value to enter a password that might be required to map the remote drive.

Here’s an example:

```vba
Set objWshNetwork = WScript.CreateObject("WScript.Network")
objWshNetwork.MapNetworkDrive "Z:","\ZEUS\SharedDocs"
```

To remove a mapped network drive, use the WshNetwork object’s RemoveNetworkDrive:

```vba
WshNetwork.RemoveNetworkDrive strName, [bForce], [bUpdateProfile]
```

- **WshNetwork**: The WshNetwork object.
- **strName**: The name of the mapped network drive you want removed. If you use a network path, all mappings to that path are removed; if you use a local drive letter, only that mapping is removed.
- **bForce**: If True, the resource is removed even if it is currently being used.
- **bUpdateProfile**: If True, the network drive mapping is removed from the user’s profile.

Here’s an example:

```vba
Set objWshNetwork = WScript.CreateObject("WScript.Network")
onjWshNetwork.RemoveNetworkDrive "Z:"
```

**Example: Scripting Internet Explorer**

To give you a taste of the power and flexibility of scripting—particularly Automation programming—I’ll close this chapter by showing you how to program a specific Automation server: Internet Explorer. You’ll see that your scripts can control just about everything associated with Internet Explorer:

- The position and dimensions of the window
- Whether the menu bar, toolbar, and status bar are displayed
- The current URL
- Sending the browser backward and forward between navigated URLs
Displaying a Web Page

To get started, I'll show you how to use the InternetExplorer object to display a specified URL. You use the Navigate method to do this, and this method uses the following syntax:

InternetExplorer.Navigate URL [, Flags,] [ TargetFrameName] [, PostData] [ ,Headers]

- **InternetExplorer**: A reference to the InternetExplorer object with which you're working.
- **URL**: The address of the web page you want to display.
- **Flags**: One of (or the sum of two or more of) the following integers that control various aspects of the navigation:
  1. Opens the URL in a new window
  2. Prevents the URL from being added to the history list
  4. Prevents the browser from reading the page from the disk cache
  8. Prevents the URL from being added to the disk cache
- **TargetFrameName**: The name of the frame in which to display the URL.
- **PostData**: Specifies additional POST information that HTTP requires to resolve the hyperlink. The most common uses for this argument are to send a web server the contents of a form, the coordinates of an imagemap, or a search parameter for an ASP file. If you leave this argument blank, this method issues a GET call.
- **Headers**: Specifies header data for the HTTP header.

Here's an example:

```vbscript
Set objIE = CreateObject("InternetExplorer.Application")
objIE.Navigate "http://www.microsoft.com/"
```

Navigating Pages

Displaying a specified web page isn’t the only thing the InternetExplorer object can do. It also has quite a few methods that give you the ability to navigate backward and forward through visited web pages, refresh the current page, stop the current download, and more. Here's a summary of these methods:

- **GoBack**: Navigates backward to a previously visited page
- **GoForward**: Navigates forward to a previously visited page
- **GoHome**: Navigates to Internet Explorer's default Home page
GoSearch
Navigates to Internet Explorer's default Search page

Refresh
Refreshes the current page

Refresh2
Refreshes the current page using the following syntax:

```
Refresh2(Level)
```

**Level** A constant that determines how the page is refreshed:

- **0** Refreshes the page with a cached copy
- **1** Refreshes the page with a cached copy only if the page has expired
- **3** Performs a full refresh (doesn't use a cached copy)

Stop
Cancels the current download or shuts down dynamic page objects, such as background sounds and animations.

**Using the InternetExplorer Object's Properties**

Here's a summary of many of the properties associated with the InternetExplorer object:

- **Busy** Returns True if the InternetExplorer object is in the process of downloading text or graphics. This property returns False when a download of the complete document has finished.

- **FullScreen** A Boolean value that toggles Internet Explorer between the normal window and a full-screen window in which the title bar, menu bar, toolbar, and status bar are hidden.

- **Height** Returns or sets the window height.

- **Left** Returns or sets the position of the left edge of the window.

- **LocationName** Returns the title of the current document.

- **LocationURL** Returns the URL of the current document.

- **MenuBar** A Boolean value that toggles the menu bar on and off.

- ** StatusBar** A Boolean value that toggles the status bar on and off.

- **StatusText** Returns or sets the status bar text.

- **ToolBar** A Boolean value that toggles the toolbar on and off.

- **Top** Returns or sets the position of the top edge of the window.

- **Type** Returns the type of document currently loaded in the browser.

- **Visible** A Boolean value that toggles the object between hidden and visible.

- **Width** Returns or sets the window width.
Running Through a Sample Script

To put some of the properties and methods into practice, Listing 12.10 shows a sample script.

LISTING 12.10 A Script That Puts the InternetExplorer Object Through Its Paces

```vbscript
Option Explicit
Dim objIE, objWshShell, strMessage, intResult

' Set up the Automation objects
Set objIE = WScript.CreateObject("InternetExplorer.Application")
Set objWshShell = WScript.CreateObject("WScript.Shell")

' Navigate to a page and customize the browser window
objIE.Navigate "http://www.wordspy.com/"
objIE.Toolbar = False
objIE.StatusBar = False
objIE.MenuBar = False

' Twiddle thumbs while the page loads
Do While objIE.Busy
Loop

' Get the page info
strMessage = "Current URL: " & objIE.LocationURL & vbCrLf & "Current Title: " & objIE.LocationName & vbCrLf & "Document Type: " & objIE.Type & vbCrLf & vbCrLf & "Would you like to view this document?"

' Display the info
intResult = objWshShell.Popup(strMessage, , "Scripting IE", vbYesNo + vbQuestion)

' Check the result
If intResult = vbYes Then

' If Yes, make browser visible
objIE.Visible = True
Else

' If no, bail out
objIE.Quit
End If
Set objIE = Nothing
Set objWshShell = Nothing
```
The script begins by creating instances of the InternetExplorer and WScript Shell objects. The Navigate method displays a page, and then turns off the toolbar, status bar, and menu bar. A Do...Loop checks the Busy property and loops while it’s True. In other words, this loop won’t exit until the page is fully loaded. A string variable is used to store the URL, the title, and type of the page, and this string is then displayed in a Popup box, which also asks whether the user wants to see the page. If the user clicks the Yes button, the browser is made visible; if the user clicks the No button, the Quit method shuts down the browser.

**From Here**

Here are some sections of the book that contain information related to the scripting techniques you learned in this chapter:

- To learn how to run scripts when you start your Windows XP system, see the section “Specifying Startup and Logon Scripts” in Chapter 5, “Installing and Running Applications.”

- To learn more about the Registry, see Chapter 11, “Getting to Know the Windows XP Registry.”

- I show you a script that displays the available free space on all your drives in the “Checking Free Disk Space” section of Chapter 15, “Maintaining Your Windows XP System.”

- For some examples of security-related scripts, see Chapter 21, “Implementing Windows XP’s Internet Security and Privacy Features.”

- You can also “program” Windows XP using batch files. See Appendix C, “Automating Windows XP with Batch Files.”
Symbols

8.3 filename creation, turning off, 397

A

accelerator keys for actions, defining, 110
accessing
Address bars, 483-484
BIOS, optimizing startup, 389
Command Prompt, 691
  adding parameters, 697-698
  CMD.EXE switches, 692, 694-695
  configuring, 699-707
  customizing, 711-712, 714, 716, 718-721, 723-724
  executing commands, 696
  sharing, 709-711
  starting applications, 707, 709
Internet, configuring firewalls, 566-569
Links bars, 502-503
networks, 631
  activating discovery, 634
  adding devices, 632
  creating locations, 638
  mapping folders, 636-638
  printing, 639
  resources, 634-635
  sharing resources, 639-643, 645-647
  troubleshooting, 633-634
  viewing devices, 632
newsgroups, 544
remote connections, 663, 665-672
  configuring, 664-665
  disconnecting, 672
Internet, 673-676
VPNs, 677-681
search engines, 490-491
configuring, 491-493
Windows Meeting Space, 655-656, 658-659
closing, 661
inviting, 657-658
joining, 656
managing, 660
sharing handouts, 658
accounts. See also user accounts
email
  backups, 537
  configuring, 516-519
faxes, creating, 235
lockout policies, setting 164
networks, sharing resources, 642-643
newsgroups, configuring, 549-550
ACPI (Advanced Configuration and Power Interface), 476
actions
  accelerator keys, defining, 110
  for file types
    changing default, 108-109
    creating new, 109-112
activating Parental Controls, 173-175
ad hoc connections, 35
ad hoc wireless networks, connecting to, 617-618
Add Hardware icon (Control Panel), 277
Add Hardware Wizard, 462
Add Network Location Wizard, 639
Add or Remove Programs. See Installed Programs dialog box
Add Printer Wizard, 639
add-ons
  Internet Explorer
    applying, 577
    managing, 583
  Manage Add-ons feature, 29
adding
  applications to Open With dialog box, 121
  calendars, subscribing, 232
  comments (REM command), 728
  devices to networks, 632
  file types to New menu, 118
  files to email, 528
  graphics to fields, 241
  information fields (fax cover pages), 240
  memory, 391
  parameters (Command Prompt), 697-698
  reminders (Windows Calendar), 225
  Run command to Start menu, 133
  search engines, 490-491
  security zones, 575
  shortcuts to Favorites folders, 494
  signatures to email, 529
  text
    ECHO command, 728-729
    fields, 241
    users
      to groups, 161
      with User Accounts dialog box, 158-159
Address bars, accessing, 483-484
Address Space Layout Randomization. See ASLR
Address toolbar, 361
addresses
  email
    communicating with, 216
    creating contact groups, 214-215
    importing contacts, 212-214
    managing contacts, 209, 211-212
    modifying, 216-219
    newsgroups, 549
    spam, 594-598
    encoding, preventing IDN spoofing, 581-583
IP, remote connections, 674-675
replies, applying different, 518
web pages, 480
AddWindowsPrinterConnection method, 344
Administrative Tools icon (Control Panel), 277
administrator
launching applications as, 139-140
opening command prompt as, 128
Administrator account, 152
including in Welcome screen, 56
Administrators group, 150, 152
Advanced Boot Options menu
customizing startup process, 52-54
when to use, 58-59
Advanced Configuration and Power Interface.
See ACPI
advanced email accounts, configuring, 518
advanced searches, 88-90
Advanced Tag Editor (Windows Media Player 11), 31, 196
Aero Glass
graphics requirements, 9
new features, 15
album art (Windows Media Player 11), 31, 194-195
alerts, incoming email messages, 520
aliases, defined, 297
All Programs menu icons, customizing, 359, 361
all-day events, creating, 227
allowing specific games, 178
Always Open With check box (Open With dialog box), disabling, 122
answering incoming calls (faxes), 243
application compatibility
compatibility layers
defined, 144-146
scripting, 145, 147-148
described, 142-143
determining, 143-144
pre-installation checklist, 125
application mixer, defined, 185
application-specific paths
creating, 140-141
in Registry, 132
applications
adding to Open With dialog box, 121
associations with multiple file types, 115
boot applications, 42
Command Prompt
sharing, 709-711
starting, 707, 709
configuring, determining causes of problems, 443
default programs, setting, 356-357
deleting, 415-417
from file types’ Open With menu, 120
from Open With dialog box, 121
file extension associations, changing, 113-114
force quitting, 443
installing
causes of problems, determining, 443-444
methods for, 130-131
pre-installation checklist, 124-130
Registry settings, 131-132
User Account Control, 124
launching
as administrator, 139-140
creating application-specific paths, 140-141
customizing AutoPlay feature, 134-135
methods of, 132-134
at startup, 134-139, 141-142
networks, sharing, 634-635, 639-643, 645-647
How can we make this index more useful? Email us at indexes@samspublishing.com
optimizing, 391-392
  adding memory, 391
  configuring priority, 393-394
  installing fast hard drives, 391
  starting, 392
unassociated applications
defined, 119
  opening documents with, 119-120
uninstalling, 126-127
WSH, 313, 315
  applying CScript, 317
  applying WScript, 316-317
  customizing properties, 318-319
  executing scripts, 315
  Internet Explorer, 345-346, 348-349
  programming objects, 319-324
  running scripts, 315
  WScript objects, 324-331
  WshNetwork objects, 343-345
  WshShell objects, 331-333, 335-341, 343
applying
  add-ons (Internet Explorer), 577
  CScript, 317
  stationary to email, 528
  tabs, 485-488
  WScript, 316-317
appointments, tracking, 222-228. See also Windows Calendar
arguments, naming, 322
articles (newsgroups), 546
as-you-type searches. See Instant Search
ASLR (Address Space Layout Randomization), 608
assigning objects to variables, 322-323
associating
  file extensions
    of applications with multiple file types, 115
changing, 113-114
  with file types, 116-117
  file types with Windows Media Player, 200
asynchronous startup, 374
attaching
  external monitors to notebook computers, 256
  second monitor, 370
attachments
  business cards, modifying contacts, 218-219
  email, 522
    forwarding, 523
  newsgroups, decoding, 555
attacks. See security
attendees (Windows Calendar), inviting, 226
audio, 184. See also digital media
  device properties, 187-188
  incoming email messages, 520
  per-application volume control, 184-186
  Sound Recorder, 186-187
audio CDs (Windows Media Player 11)
  burning, 207-208
  ripping, 204-207
audio stack, defined, 184
authentication
  rules, 569
  SMTP, enabling, 518
AUTOEXEC.BAT file, 292
Automatic Learning (handwriting recognition), 263
automatic logon, 57, 391
automating
  backups, 425, 427
  Batch files, 725
    commands, 727-729
    creating, 726
    GOTO command, 739-740
IF command, 741-744
loops, 735-739
parameters, 729-735
piping commands, 749-750
redirecting, 747-749
saving, 726-727
XCOPY command, 745-746
Disk Defragmenter, 418
hardware installations, 463
scripts, 325-331
AutoPlay feature
customizing, 134-135, 182-183
installing applications, 130
launching applications, 133
AutoPlay icon (Control Panel), 278
AutoRun Registry settings, 693
auxiliary displays. See secondary displays

B
customizing startup process, 41-42
creating custom, 253-255
list of, 252-253
monitoring battery life, 251-252
battery status setting (Windows Mobility Center), 249
BCD (Boot Configuration Data), 40
customizing startup process, 41-42
with BCDEDIT, 47-50, 52
with Startup and Recovery dialog box, 43-44
with System Configuration Utility, 44-47
BCD store, backing up, 50
BCDEDIT, customizing BCD options, 47-50, 52
benchmarks, collector sets, 389
binary values in Registry, editing, 306-307

How can we make this index more useful? Email us at indexes@samspublishing.com
BIOS

checks, reducing, 389
Plug and Play devices, installing, 458
settings, adjusting, 60, 454

bit buckets, 748
bit rate, defined, 206
BitLocker Drive Encryption icon (Control Panel), 279

blocking
pop-ups, 587-588
read receipts, 599
senders (email), 597
specific games, 178

blue screen of death (BSOD). See system failures
Bluetooth Devices icon (Control Panel), 279
bodies, retrieving news group message, 554
boot applications, 42
boot code, 40
Boot Configuration Data (BCD), 40
    customizing startup process, 41-42
    with BCDEdit, 47-50, 52
    with Startup and Recovery dialog box, 43-44
    with System Configuration Utility, 44-47
boot process. See startup process
boot sectors, 40
BOOT.INI file, 42
booting from last known good configuration, 452
BOOTMGR (Windows Boot Manager), 40
breadcrumming, defined, 68-69
brightness setting (Windows Mobility Center), 249
broadband connections, configuring firewalls, 566-569
BSOD (blue screen of death). See system failures
buffers, overflow, 607
bugs
origin of term, 438
web, 599-600
burning
audio CDs (Windows Media Player 11), 207-208
DVDs, new features, 31, 33
business cards, modifying contacts, 218-219

cache memory, 6
caching
enabling, 395-396
web pages, 504-505
calendars, scheduling, 219-220.
See also Windows Calendar
    configuring multiple calendars, 229
importing files, 230
modifying views, 221-222
navigating dates, 220
sharing, 231-233
tracking appointments, 222-228
cameras, installing, 462
canceling
    drag-and-drop, 78
    I/O, 404
    newsgroup messages, 555
categories, selecting (Windows Media Player 11), 193-194
CD command, 733
CDA format, 206
CDOs (Collection Data Objects), 592
CDs
audio CDs. See audio CDs
AutoPlay feature. See AutoPlay feature
booting from, 60
Change Network Icon dialog box, 621
changing. See also customizing
default actions for file types, 108-109
file extension associations, 113-114
channels, email, 600
charts, System Stability Chart, 405
check boxes, selecting files, 74-75
Check Disk, 409
running, 410-412
checking hard disks for errors, 407-412
checklists
causes of problems, determining, 438-444
pre-installation checklist, 124-130
Choose Details dialog box, 97
Choose Search Locations dialog box, 88
classic menus, viewing/hiding, 71
Classic View (Control Panel), 277
clearing recent programs list (Start menu), 354
ClearType, 370
clients (VPNs), configuring, 679-680
clocks, viewing multiple time zones on taskbar, 365-367
closing
8.3 filename creation, 397
Indexers, 395
scripts, 325
splash screens, 390
Windows Meeting Space sessions, 661
clusters, 408-409
CMD.EXE switches, 692, 694-695
codenames (Microsoft), 2
collaborating with Windows Meeting Space, 654
closing, 661
inviting, 657-658
joining, 656
managing, 660
sharing handouts, 658
signing in, 655
starting, 655-656, 658-659
Collection Data Objects. See CDOs
collections, objects, 323-324
Color Management icon (Control Panel), 279
color quality
coloring, 370-371
defined, 369-370
colors, Command Prompt, 715
columns (email), customizing, 524
command line
Check Disk, running, 411-412
Disk Fragmenter, 421
optimizing, 732-734
Command Prompt, 61, 455, 691
applications
sharing, 709-711
starting, 707, 709
CMD.EXE switches, 692, 694-695
commands
adding parameters, 697-698
executing, 696
configuring, 699-707
customizing, 711-712, 714, 716, 718-721, 723-724
.exe file extension at, 141
folders, mapping, 637
NET LOCALGROUP command, 166
NET USER command, 165-166
opening
as administrator, 128
in current folder, 111-112
starting session, 165
commands. See also Command Prompt
Batch files, 727
CD, 733
DEL, 732-733
ECHO, 728-729
FOR, 735-739
GOTO, 739-740
IF, 741-744
parameters, 729-735
PAUSE, 729
piping, 749-750
REM, 728
SHIFT, 734-735
XCOPY, 733-734, 745-746
DEFRA G, 421
input, redirecting, 748-749
My Business Card, 219
output, redirecting, 747
PATH, 726-727
SORT, 749
undoing, 81
comments
in .reg files, 308
REM command, 728
commercial news servers, 545
commit limit, defined, 447
comparing text files, 128-129
compatibility
of data files, pre-installation checklist, 126
hardware, 458
with Windows Vista
compatibility layers, 144-148
described, 142-143
determining, 143-144
pre-installation checklist, 125
compatibility layers
defined, 144-146
scripting, 145, 147-148
CompletePC backup, new features, 34
compression
disabling, 395
sound quality versus, 206
Computer Management icon (Control Panel),
277
computer names, changing, 611-612
computer problems. See problems, troubleshooting
computers, sharing, 179
conditions, IF command, 741-744
CONFIG.SYS file, 292
configuration files, history of, 292-293
configuring
accounts, newsgroups, 549-550
ACPI, 476
applications, determining causes of problems, 443
automatic logon, 391
backups
automating, 425, 427
creating system images, 427
Batch files, 726
commands, 727-729
GOTO command, 739-740
IF command, 741-744
loops, 735-739
parameters, 729-735
piping commands, 749-750
redirecting, 747-749
saving, 726-727
XCOPY command, 745-746
Command Prompt, 699
formatting long filenames, 699
modifying folders, 700
optimizing DOSKEY, 701-707
sharing applications, 709-711
starting applications, 707, 709
Control Panel with group policies, 289-290
data collector sets, 389
Disk Defragmenter, 418
email
  accounts, 516-519
  customizing columns, 524
  filtering, 537-538, 540
  incoming messages, 520-523
  managing, 535-537
  optimizing privacy, 599-600
  phishing filters, 598
  reading, 525-527
  receiving, 600-604
  Safe Senders, 596
  searching, 540-542
  sending, 527-534, 600-604
faxes, 233
  creating accounts, 235
  formatting cover pages, 235, 239-241
  receiving, 242-244
  sending, 236-238
  starting Windows Fax and Scan, 234-235
firewalls, 566-569
Internet Explorer
  customizing options, 505, 507, 509-514
  modifying home pages, 505-506
keyboard shortcuts, 683-689
Last Access Time, 397
multiple calendars, 229
networks
  creating locations, 638
  printing, 639
power buttons (notebook computers), 255-256
prefetchers, 391
priority, 393-394
Problem Reports and Solutions, 449-450
properties, values, 320
Registry
  AutoRun, 693
  deleting settings, 341
  storing settings, 340
  remote connections, 664-665
search engines, 491-493
security
  adding removing zone sites, 575
  applying add-ons, 577
  blocking pop-ups, 587-588
  Control Panel, 564
  customizing, 589-590
  deleting browser history, 584-585
  email, 590. See also email
  encoding addresses, 581-583
  Internet Explorer, 573-574
  managing add-ons, 583
  modifying zone sites, 576
  new features, 565-566
  optimizing privacy, 585-587
  phishing filters, 578-580
  policies, 470-471
  reducing privileges, 576-577
  remote connections, 674
shortcuts, 337
system failure handling, 440-442
System Restore, 421-423
VPNs
  clients, 679-680
  gateways, 678-679
Windows, determining causes of problems, 440
Windows Defender, 572-573
Windows Media Player security, 605-606

How can we make this index more useful? Email us at indexes@samspublishing.com
conflicts (resources), troubleshooting, 476-477

Connect to a Network dialog box, 613

connections. See also networking

activating discovery, 634
to ad hoc wireless networks, 617-618
broadband, configuring firewalls, 566-569
Network Connections dialog box, 625-626
networks, 631

adding devices, 632
creating locations, 638
mapping folders, 636-638
printing, 639
resources, 634-635
sharing resources, 639-643, 645-647
troubleshooting, 633-634
viewing devices, 632
to nonbroadcasting wireless networks, 615-616
remote, 663, 665-672
configuring, 664-665
disconnecting, 672
Internet, 673-676
VPNs, 677-681
security, rules, 569
to wireless networks, 612-615

contacts

communicating with, 216
groups, creating, 214-215
importing, 212-214
managing, 209, 211-212
modifying, 216-219

Contacts folders, 210

Control Panel

configuring with group policies, 289-290
CPL files, opening, 282-284
icons
alternate access to, 284-285
list of, 277-282

new features, 24, 26-27
security, configuring, 564
on Start menu, converting icons to menu, 285-286
viewing, 275-277

control sets, defined, 452

CONTROL.INI file, 292

colors, managing, 585-587

Copy File dialog box, 76

copying

files
resolving name conflicts, 76-77
Send To command, 78-79
GUIDs (globally unique identifiers), 51
music to/from CDs (Windows Media Player 11), 204-208
text (Command Prompt), 709-710
XCOPY commands, 733-734

countries, blocking, 597

Cover Page Editor Drawing toolbar, 241

covers (faxes)
formatting, 235, 239-241
sending, 237

CPL files, opening, 282-284

crashes. See stability; system failures; troubleshooting

CreateObject method, 326

CreateShortcut method, 337

Cryptographic Operators group, 150

CScript, applying, 317

Ctrl+Alt+Delete key combination, requiring for logon process, 54-55
current folder, opening command prompt in, 111-112

Custom install option (pre-installation checklist), 130

custom power plans (notebook computers), creating, 253-255

Custom Scan, Windows Defender, 571

Customize Start Menu dialog box, 352

customizing

AutoPlay, 134-135, 182-183
Command Prompt, 711-712, 714, 716, 718-721, 723-724
e-mail

columns, 524
reading, 525-527
sending, 531-534
environment variables, 738-739
faxes

receiving, 242-244
sending, 236-238
Internet Explorer, 502

accessing Links bars, 502-503
caching, 504-505
configuring options, 505, 507, 509-514
modifying home pages, 505-506
logon process

automatic logon, 57
including Administrator account in Welcome screen, 56
requiring Ctrl+Alt+Delete for, 54-55
maintenance, scheduling, 434-435
networks, 621-622
New menu, 117-118
newsgroups, 559-562
Open With dialog box, 118

adding applications to, 121
deleting applications from, 121
deleting applications from file types menu, 120
disabling Always Open With check box, 122

opening documents with unassociated applications, 119-120
Registry settings, 120
page files, 398-399
screen resolution and color quality, 370-371
scripts, properties, 318-319
security, 589-590

policies, 470-471
zones, 576
spyware scanning, 572-573

Start menu, 352

All Programs menu icons, 359, 361
clearing recent programs list, 354
converting links to menus, 358-359
default programs, setting, 356-357
favorite programs area, 352-354
with group policies, 368-369
Internet and E-mail icons, 355-356
pinned programs list, 357-358

startup process

with Advanced Boot Options menu, 52-54
with BCD (Boot Configuration Data), 41-50, 52
taskbar

creating toolbars, 362
customizing toolbars, 362
with group policies, 368-369
grouping taskbar buttons, 367-368
setting options, 363-365
viewing multiple time zones, 365-367
viewing toolbars, 361-362
Windows Error Reporting service, 404
Windows Explorer

changing views, 95
details view, 96-97

How can we make this index more useful? Email us at indexes@samspublishing.com
full-screen mode, 97
moving user folders, 101-102
view options, list of, 98-101
Windows Media Player security, 605-606

cycles, 409
configuring, 393-394

data collector sets, 388
data files, compatibility (pre-installation check-list), 126. See also documents
data source names, defined, 278
Data Sources (ODBC) icon (Control Panel), 278
Date and Time dialog box, 365
Date and time icon (Control Panel), 279, 284
dates, navigating, 220
deactivating Sharing Wizard, 641-642
debugging. See troubleshooting
decoding messages (newsgroups), 555
decompressing downloaded files, 130
default actions for file types, changing, 108-109
default email accounts, configuring, 517
default programs, setting, 356-357
Default Programs icon (Control Panel), 279, 285
defining accelerator keys for actions, 110
defragmenting hard disks, 417-419
DEL command, 732-733
delayed environment variable expansions, 738-739
Delete Browsing History option (Internet Explorer), 28
deleting
Address bar lists, 483
applications
from file types’ Open With menu, 120
from Open With dialog box, 121
custom power plans (notebook computers), 254
e-mail messages, 523
file types from New menu, 118
files, 415-417
recovery of deleted files, 81
Recycle Bin, 79-81
histories (Internet Explorer), 584-585
mapped network drives, 345
Registry keys, 310
Registry settings, 341
security zones, 575
tasks, 229
desktop, new features, 15
Desktop toolbar, 362
Desktop Window Manager (DWM), new features, 23
Details view (Windows Explorer), 71-72, 96-97
device driver installation, determining causes of problems, 444
Device Manager, 457-458
devices, installing, 458
hardware, 465-468, 470
legacy devices, installing, 462-463
Plug and Play devices, installing, 458-460, 462
security policies, 470-471
signing options, 463-465
troubleshooting, 472-477
Device Manager icon (Control Panel), 279
devices
audio devices, properties, 187-188
drivers, upgrading, 390, 392
installing, determining causes of problems, 444
networks
adding, 632
viewing, 632
encryption, 395

games, 177

Last Access Time, 397

network connections, 626

network discovery, 622-623

password requirements for Sleep mode, 256

Registry Editor access, 300

SSID broadcasting, 615

unsigned drivers, 474

Welcome Center, 281

Windows Media Player updates, 202

disconnecting. See also accessing; connections

mapped network folders, 638

remote connections, 672

discovery (networks), activating, 634

discussion topics, newsgroups, 546

Disk Cleanup utility, 415

Disk Diagnostics, 21

disk space. See storage requirements

disks. See hard disks

Display Settings dialog box, 370

displaying Windows Boot Manager manually, 52. See also viewing

Distributed COM Users group, 150

Distributed File System Replicator. See DFSR

DLL (dynamic link library), 313

DNS (domain name system), 566

document-specific paths, creating, 141

documentation, pre-installation checklist, 125

documents. See also data files; faxes; text files

keyboard shortcuts, 684

opening with unassociated applications, 119-120

Documents window, 66-67

domain name system. See DNS

domain networks, defined, 621

domains

IDNs, 579

preventing spoofing, 581-583

logging on, 55-56

DOSKEY

Command Prompt, 701-707

keyboard shortcuts, 688

double-pumped memory, 7

down-level tools for Windows XP, 4

downloaded files, decompressing, 130

downloading

drivers, 475

messages (newsgroups), 553-554

DPS (Diagnostic Policy Service), 446

dragging/dropping

shortcuts onto Start button, 360

tips for, 77-78

drawing (Cover Page Editor Drawing toolbar), 241

drive letters, hiding/viewing, 100

drivers. See also devices

devices, troubleshooting, 473

downloading, 475

reinstalling, 474

rolling back, 474

updating, 467-468, 470

upgrading, 390, 392

drives (networks), mapping, 344-345

dual-core processors, 6

Dumpster diving, defined, 168

DVD burners, 8

DVDs

AutoPlay feature. See AutoPlay feature

booting from, troubleshooting, 454

burning, 31, 33

playing, 32

DWM (Desktop Window Manager), 23
DWORD values (in Registry), editing, 305-306
Dynamic Host Control Protocol (DHCP), 610
dynamic link library. See DLL

E
E-mail icon (Start menu), customizing, 355-356
Ease of Access Center icon (Control Panel), 279
ECHO command, 728-729
Edit String dialog box, 109-110
editing. See also customizing
  command lines, 702
cover pages, 240
images (Windows Photo Gallery), 191-192
metadata, 83-84
Registry. See Registry Editor
user accounts, 156-158
editions
  of Windows Vista, 10-12
    upgrading with Windows Anytime Upgrade feature, 12-14
  of Windows XP 10
electronic business cards, modifying contacts, 218-219
elements (collections), 323-324
elevating privileges, 153-154
eliminating BIOS checks, 389
Elliot, James, 544
email. See also Windows Mail
  accounts
    backups, 537
    configuring, 516-519
contacts
  communicating with, 216
  creating groups, 214-215
  importing, 212-214
  managing, 209, 211-212
  modifying, 216-219
messages
  customizing columns, 524
  filtering, 537-538, 540
  incoming, 520-521
  managing, 535-537
  processing, 522-523
  reading, 525-527
  searching, 540-542
  sending, 527-534
newsgroups, 549
security, 590
  optimizing privacy, 599-600
  phishing filters, 578-580, 598
  sending/receiving, 600-604
  spam protection, 594-598
  virus protection, 591, 593-594
Trojan horses, 567
enabling
  network discovery, 622-623
  offline files, 647-650
  SMTP authentication, 518
  write caching, 395-396
encoding addresses, preventing IDN spoofing, 581-583
encryption
  disabling, 395
  email, 529. See also email
  reversible encryption, 170
ending. See quitting
engines, scripting, 313, 315
entries
  Registry. See also keys (Registry)
    changing values of, 303, 305-307, 309
    finding, 310-311
  Windows Boot Manager, 51
environments
delayed variable expansions, 738-739
variables, WshShell objects, 341, 343
   WSH. See scripts; WSH
error handling. See troubleshooting
error messages, determining causes of problems, 438-439
Error Reporting dialog box, 404
errors. See also troubleshooting
   Event Viewer, 430, 432, 434
   hard disks
      checking for, 407-409
      running Check Disk, 410-412
etiquette rules, newsgroups, 547-548
Event Viewer, 430, 432, 434
   logs, determining causes of problems, 439
Event Viewer icon (Control Panel), 278
events
   all-day, creating, 227
   defined, 278
   Reliability Monitor, 405
.exe file extension in Run dialog box/command prompt, 141
executing. See also launching; opening; running
   Check Disk, 410-412
   commands (Command Prompt), 696
   Disk Defragmenter, 417-419
   scripts, 315
   System Restore, 421-423
exit code, 745
exporting Registry keys, 301-303
exposing objects, 329-330
extensions. See file extensions
external display setting (Windows Mobility Center), 250
external monitors, attaching to notebook computers, 256

F
failures. See system failures
FAT (file allocation table) file system, partitions, 143
Favorite Links section (folders), 73
favorite programs area (Start menu), customizing, 352-354
favorites, saving Registry keys as, 304
Favorites folders, 493
   shortcuts
      adding, 494
      managing, 495-496
      opening, 494-495
      sharing, 496-497
Fax Accounts dialog box, 235
Fax Settings dialog box, 236
Fax Setup Wizard, 235
faxes
   configuring, 233
   creating accounts, 235
   formatting cover pages, 235, 239-241
   starting Windows Fax and Scan, 234-235
   receiving, 242-244
   sending, 236-238
FC (File Compare) command, 128-129
fields, adding graphics/text to, 241
file and Registry virtualization, defined, 154
file extensions. See also file types
   associating with file types, 116-117
   changing associations, 113-114
   file types and, 104-106
   hiding, 100
   permanently, 112
   problems with, 104-105
   registering, 115-116
   viewing, 100, 105
file formats (Windows Media Player 11), 204, 206

file systems
- clusters, 408-409
- FAT, partitions, 143
- NTFS, partitions, 143
- TxF (Transactional NTFS), 93-94
- WinFS, 65

File Transfer Protocol. See FTP

file types. See also file extensions
- actions, creating new, 109-112
- adding to New menu, 118
- associating
  - file extensions with, 116-117
  - multiple file types with applications, 115
  - with Windows Media Player, 200
- creating, 115-116
- default actions, changing, 108-109
- deleting from New menu, 118
- file extensions and, 104-106
- hiding file extensions permanently, 112
- Open With menu, deleting applications from, 120
- Registry and, 106-108, 132
  - ShellNew subkey, 117-118

files. See also data files; documents; folders;
text files; Windows Explorer
- backups, 424
  - automating, 425, 427
- Batch
  - commands, 727-729
  - creating, 726
  - GOTO command, 739-740
  - IF command, 741-744
  - loops, 735-739
  - overview of, 725
  - parameters, 729-735
  - piping commands, 749-750
- redirecting, 747-749
- saving, 726-727
- XCOPY command, 745-746
- calendars, importing, 230
- configuration files, history of, 292-293
- copying, resolving name conflicts, 76-77
- deleting, 415-417
  - recovering deleted files, 81
  - Recycle Bin, 79-81
- Disk Defragmenter, 417-419
- drag-and-drop tips, 77-78
- email, adding, 528
- filtering, 19, 92-93
- grouping, 18, 91-92
- hidden files, viewing, 99
- long filenames, 699
- maintaining with Open and Save As dialog boxes, 82
- media files, playing in Windows Media Player 11, 200-201
- moving, resolving name conflicts, 76-77
- networks, sharing, 634-635, 639-643, 645-647
- offline, 647
  - enabling, 647-650
  - modifying disk space, 653-654
  - synchronizing, 650-653
- page
  - customizing, 398-399
  - modifying, 399-401
  - monitoring, 399
  - saving, 398
  - splitting, 398
- PIFs, 698
- Registry
  - hives, 299-300
  - WshShell objects, 339-341
- renaming with hidden file extensions, 100
reverting to previous versions, 94-95
ripping, 31
scripts
  applying CScript, 317
  applying WScript, 316-317
  customizing properties, 318-319
  executing, 315
  running, 315
searching. See searches
  selecting
    with check boxes, 74-75
    lassoing, 77
Send To command, 78-79
sharing (Public folder), 171-172
shortcuts (WshShell objects), 336-339
Size value versus Size on Disk value, 75
stacking, 18, 92
System Restore, 421-423
Files and Settings Transfer Wizard. See
Windows Easy Transfer
FileSystemObject, programming, 330-331
filtering
  email, 537-538, 540
    spam, 594-598
  files, 19, 92-93
  messages (newsgroups), 557
  phishing, 578-580, 598
Find Message dialog box, 541-542
finding
  Registry changes, 302
  Registry entries, 310-311
firewalls. See also Windows Firewall
  configuring, 566-569
  remote connections, 674
flash memory, 250
Flip 3D feature, 16
Flip feature, 16
Folder Options icon (Control Panel), 279, 284
folder view, filtering, 19
folder windows, new features, 16
folders. See also files; Windows Explorer
  Batch files, 727
  Command Prompt, modifying, 700
  Contacts, 210
  current folder, opening command prompt in, 111-112
  Details pane, 71-72
  email, moving messages, 523
  Favorite Links section, 73
  Favorites, 493
    adding shortcuts, 494
    managing shortcuts, 495-496
    opening shortcuts, 494-495
    sharing shortcuts, 496-497
  installation folders, pre-installation checklist, 129
  Instant Search box, 69-70
  in interface, 66-67
  keyboard shortcuts, 686
  Live Icons, 73-74
  navigation, 67-69
  Navigation pane, 73
  networks
    mapping, 636-638
    sharing, 634-635, 639-643, 645-647
  Preview pane, 72
  Public folder, 171-172
  reverting to previous versions, 94-95
  Size value versus Size on Disk value, 75
  Start Menu folder, customizing shortcuts, 360
  Startup folder, 134-135
  Task pane, 70
  user folders, moving, 101-102
Folders list, viewing/hiding, 73
fonts
  Command Prompt, modifying, 713
  sharpening with ClearType, 370
Fonts icon (Control Panel), 279, 284
FOR command, 735-739
For Each…Next loops, 323
for loops, 324
For…Next loops, 323
force quitting applications, 443
foreground programs. See applications
forgotten passwords, recovering, 170-171
formatting
  Batch files, 726
    commands, 727-729
    GOTO command, 739-740
    IF command, 741-744
    loops, 735-739
    parameters, 729-735
    piping commands, 749-750
    redirecting, 747-749
    saving, 726-727
    XCOPY command, 745-746
Command Prompt, customizing, 714
contacts, 211-212
  communicating with, 216
  creating groups, 214-215
  importing, 212-214
  modifying, 216-219
cover pages (faxes), 235-241
e-mail
  applying recipient shortcuts, 530
  customizing, 531-534
  signatures, 530
Favorites folders, 493
  adding shortcuts, 494
  managing shortcuts, 495-496
  opening shortcuts, 494-495
  sharing shortcuts, 496-497
long filenames, 699
partitions, clusters, 408-409
scripts
  applying CScript, 317
  applying WScript, 316-317
  customizing properties, 318-319
  executing, 315
  running, 315
  shortcuts, 337
  URLs, creating shortcuts, 484-485
forwarding
  email messages, 523
  ports, 676
free disk space, checking, 412-414
freeze-drying, defined, 379
freezes. See stability; troubleshooting; system failures
FTP (File Transfer Protocol), 566
Full Scan (Windows Defender), 571
full-screen mode (Windows Explorer), 97

g
Game Controllers icon (Control Panel), 279
Game Explorer, new features, 34-35
games
  blocking/allowing specific games, 178
  disabling, 177
  Game Explorer, new features, 34-35
  Parental Controls example, 175-178
  ratings, 177-178
Gates, Bill, 2
gateways (VPNs), configuring, 678-679
GetObject method, 328-329
global user accounts, defined, 149
globally unique identifiers (GUIDs), 50-51

How can we make this index more useful? Email us at indexes@samspublishing.com
GOTO command, 739-740

graphical user interface (GUI). See interface graphics. See also images
  Command Prompt, sharing, 711
  fields, adding, 241
  new features, 23

graphics requirements (Windows Vista), 8-9

group policies, 286
  configuring Control Panel, 289-290
  customizing Start menu/taskbar, 368-369
  devices, applying security, 470-471
  Group Policy editor, 286-289
  launching applications at startup, 136-138
  User Account Control, 155-156

Group Policy editor, 286-289

grouping
  files, 18, 91-92
  images (Windows Photo Gallery), 189-190
  media, new features, 31
  taskbar buttons, 367-368
  in Windows Media Player 11, 194, 196

groups
  contacts, creating, 214-215
  security groups, 149-151
  users, adding, 161

Guests group, 150

GUI (graphical user interface). See interface

GUIDs (globally unique identifiers), 50-51

H

hackers, firewalls. See firewalls

HAL (hardware abstraction layer), 46

handles, defined, 295

handwriting recognition (Tablet PCs), personalizing, 268-271

hard disk space. See storage requirements

hard disks
  defragmenting, 417-419
disk diagnostics, 446
  errors
    checking for, 407-409
    running Check Disk, 410-412
  free space, checking, 412-414

hard drives
  configuring Last Access Time, 397
  converting partitions to NTFS, 396-397
  installing, 391
  optimizing, 394
  closing Indexers, 395
  disabling compression/encryption, 395
  enabling write caching, 395-396
  implementing maintenance, 394
  viewing specifications, 394
  turning off 8.3 filename creation, 397

hardware
  Add Hardware Wizard, 462
  compatibility, 458
  managing, 465-468, 470

hardware abstraction layer (HAL), 46

hardware requirements (Windows Vista), 9-10

headers (messages), downloading, 553

Help and Support Center, 445

Hibernate mode, 375

hidden files, viewing, 99

hidden user profiles, 151

hiding
  classic menus, 71
  drive letters, 100
  file extensions, 100
  permanently, 112
  problems with, 104-105
Folders list, 73
notification area (taskbar) icons, 365
resources (networks), 645-647

hierarchies, 544

hints. See password hints

histories
Address bars, deleting, 483
Internet Explorer, deleting, 584-585

history
of configuration files, 292-293
of Registry, 293
of Windows releases, 1-2
of Windows Vista, 2-3

History lists, viewing, 488, 490, 506
hives (Registry), 299-300
HKEY_CLASSES_ROOT key (Registry), 106-107, 296-297
HKEY_CURRENT_CONFIG key (Registry), 298
HKEY_CURRENT_USER key (Registry), 297-298
HKEY_LOCAL_MACHINE key (Registry), 298
HKEY_USERS key (Registry), 298

Home page view (Control Panel), 276
home pages (Internet Explorer). See also Internet Explorer; web pages
modifying, 505-506
multiple home pages, 28

homograph spoofing, 579

Hopper, Grace, 438

hosts (IP addresses), remote connections, 674-675
hot swapping devices, 459
hotspots, defined, 613
hybrid hard drives, defined, 250

I/O cancellation, 20, 404

iCalendar files, importing, 230
icons
in Control Panel
alternate access to, 284-285
converting to menu on Start menu, 285-286
list of, 277-282
Live Icons, 73-74
Windows Explorer, customizing views, 95
idle sensitivity, 723-724
IDNs (Internationalized Domain Names), 579
spoofing, preventing, 581-583
IF command, 741-744
IIS_IUSRS group, 150
images. See also graphics
contacts, modifying, 217
system images, creating, 427
Windows Photo Gallery, 188-189, 191-192
applying tags, 189-190
editing images, 191-192
grouping images, 189-190
new features, 33
searching images, 191
implementing maintenance, hard drives, 394
importing
contacts, 212-214
files, calendars, 230
.reg files, 303
inbound rules, Windows Firewall, 568
inboxes. See email
incoming calls, answering, 243
incoming messages
email, 520-527
filtering, 537-538, 540

How can we make this index more useful? Email us at indexes@samspublishing.com
Indexers, closing, 395
indexes, Windows Search Engine, 85-86
Indexing Options icon (Control Panel), 279
.inf files, installing from, 131
information fields (fax cover pages), adding, 240
Infrared icon (Control Panel), 279
initialization files, 292
input commands, redirecting, 748-749
Input Panel, Tablet PCs, 263-265
InsertBefore method, 327
installation folders, pre-installation checklist, 129
Installed Programs dialog box, uninstalling applications, 126-127
installing
applications
causes of problems, determining, 443-444
methods for, 130-131
pre-installation checklist, 124-130
Registry settings, 131-132
User Account Control, 124
device drivers, determining causes of problems, 444
devices, 458
causes of problems, determining, 444
legacy, 462-463
Plug and Play, 458-460, 462
signing options, 463-465
hard drives, 391
Instant Search box, 69-70
new features, 17, 31
Start menu, 86-88
Windows Mail, 540-541
Windows Media Player 11, 197
Windows Photo Gallery, 191
interfaces. See also Internet Explorer
ACPI, 476
Check Disk, running, 410-411
customizing Start menu. See Start menu folders, 66-67
details pane, 71-72
Favorite Links section, 73
Instant Search box, 69-70
Live Icons, 73-74
navigation, 67-69
Navigation pane, 73
Preview pane, 72
Task pane, 70
new features, 14-18
Windows Media Player 11, 192-193
album art, 194-195
Library navigation, 193-194
Media Sharing, 199-200
metadata, 195, 197
searching, 197
syncing with media devices, 198-199
views, 194, 196
Internationalized Domain Names. See IDNs
Internet. See also online resources
newsgroups
articles/threads, 546
configuring accounts, 549-550
customizing, 559-562
downloading messages, 553-554
etiquette rules, 547-548
filtering messages, 557
managing messages, 555
navigating, 546
overview of Usenet, 544-545
posting messages, 556-557
rating posts, 558-559
replying to messages, 555-556
subscribing, 550-552
remote connections, 673-676

Internet Explorer
customizing, 502
accessing Links bars, 502-503
caching, 504-505
configuring options, 505, 507, 509-514
modifying home pages, 505-506
Favorites folders
adding shortcuts, 494
formatting, 493
managing shortcuts, 495-496
opening shortcuts, 494-495
sharing shortcuts, 496-497
keyboard shortcuts, 686-687
navigating
accessing Address bars, 483-484
applying tabs, 485-488
creating URL shortcuts, 484-485
optimizing, 480-482
viewing History lists, 488, 490, 506
new features, 27-29
object properties, 347
Phishing Filter, new features, 22
Protected mode, new features, 22
RSS feeds, 497-499, 501
subscribing, 499
updating, 501-502
scripting, 345-346, 348-349
searching, 490
adding search engines, 490-491
configuring search engines, 491-493
security
adding/removing zone sites, 575
applying add-ons, 577
blocking pop-ups, 587-588
configuring, 573-574
customizing, 589-590
deleting browser history, 584-585
encoding addresses, 581-583
managing add-ons, 583
modifying zone sites, 576
optimizing privacy, 585-587
phishing filters, 578-580
reducing privileges, 576-577
web page addresses, 480
in Windows XP, 4
Internet icon (Start menu), customizing, 355-356
Internet Options dialog box, 505, 574
Internet Options icon (Control Panel), 279, 284
invalid clusters. See clusters; hard disks
Invite People dialog box, 657
inviting
attendees (Windows Calendar), 226
Windows Meeting Space, 657-658
IP addresses
conflicts, troubleshooting, 611
remote connections, 674-675
iSCSI Initiator icon (Control Panel), 278-279
ISP news servers, 545
issues. See bugs; problems; system failures; troubleshooting

J

JavaScript
objects, exposing, 329-330
WSH, 313, 315
jobs, scripts, 317
joining Windows Meeting Space, 656
Junk Filter, 594-598
Junk Mail Filter, 22, 29
kernel, defined, 442
Keyboard icon (Control Panel), 280
keyboard shortcuts, 683-689
keys (public), sending/receiving secure email, 602
keys (Registry). See also entries (Registry)
   creating, 310
deleting, 310
exporting, 301-303
renaming, 309
saving as favorites, 304
Keys pane (Registry Editor), 295

labels, 740
languages, blocking, 597
LANs, remote connections, 673-676
laptop computers. See notebook computers
lassoing, defined, 77
Last Access Time, disabling, 397
last known good configuration, booting from, 452
launching. See also executing; opening; running applications
   as administrator, 139-140
creating application-specific paths, 140-141
   customizing AutoPlay feature, 134-135
   methods of, 132-134
   at startup, 134-139, 141-142
Control Panel icons, 284-285
   scripts at startup, 134-139
layouts (Command Prompt), customizing, 714
least privileged user, defined, 152
legacy applications. See compatibility
legacy devices, installing, 462-463
legacy keys, defined, 136
libraries, macros, 707
Library (Windows Media Player), navigating, 193-194
library sharing (Windows Media Player 11), new features, 32
limited user accounts, defined, 152
links (Start menu), converting to menus, 358-359
Links bars, accessing, 502-503
Links toolbar, 361
listening ports, modifying, 673
lists
   Address bars, deleting, 483
   History, viewing, 488, 490, 506
Live Icons, 73-74
Local Security Policy editor, 288
Local Security Policy icon (Control Panel), 278
local user accounts, defined, 149
Local Users and Groups snap-in, 160-161
locations (networks), creating, 638
locked out, defined, 164
logon process, 54-55. See also startup process
   automatic logon, configuring, 391
   customizing
      automatic logon, 57
      including Administrator account in Welcome screen, 56
      requiring Ctrl+Alt+Delete for, 54-55
   domain logon, 55-56
logon scripts, running, 138
logs (Event Viewer), 430, 432, 434
   causes of problems, determining, 439
Longhorn, history of Windows Vista, 2-3
lookalike attacks, 579
loops
Batch files, 735-739
for, 324
For Each...Next, 323
For...Next, 323

lossy compression, defined, 206
lost clusters. See clusters; hard disks

M
macros, DOSKEY, 703-707
Mail. See Windows Mail
mainstream hierarchies, 544
maintenance
backups, 424
automating, 425, 427
creating system images, 427
e-mail, 535-537
Event Viewer, 430, 432, 434
files, deleting, 415-417
free disk space, checking, 412-414
hard disks
checking for errors, 407-409
defragmenting, 417-419
running Check Disk, 410-412
hard drives, implementing, 394
newsgroups, 560
scheduling, 434-435
stability improvements, 403
I/O cancellation, 404
Reliability Monitor, 405
service recovery, 406
Startup Repair Tool, 407
System Restore, configuring, 421-423
updates, 428-429
checking for security vulnerabilities, 430-431

malware. See also security
ASLR, 608
Windows Service Hardening, 606
Manage Add-Ons dialog box, 29, 583
managing
contacts, 209, 211-212
communicating with, 216
creating groups, 214-215
importing, 212-214
modifying, 216-219
Device Manager, 457-458
hardware, 465-468, 470
installing devices, 458
legacy devices, 462-463
Plug and Play devices, 458-460, 462
security policies, 470-471
signing options, 463-465
troubleshooting, 472-477
e-mail, 522-523, 535-537
customizing columns, 524
optimizing privacy, 599-600
reading, 525-527
receiving, 600-604
sending, 527-534, 600-604
e-mail, optimizing privacy, 599
Internet Explorer, caching, 504-505
messages (newsgroups), 555
security
add-ons (Internet Explorer), 583
blocking pop-ups, 587-588
Control Panel, 564
customizing security, 589-590
deleting browser history, 584-585
new features, 565-566
optimizing privacy, 585-587
shortcuts, Favorites folders, 495-496
Windows Meeting Space, 660

How can we make this index more useful? Email us at indexes@samspublishing.com
manually displaying Windows Boot Manager, 52
mapping
   folders to networks, 636-638
   networks
      drives, 344-345
      printers, 344
marking
   email messages as read, 525
   newsgroup messages for downloading, 553
MBR (Master Boot Record), 40
media, grouping/stacking, 31. See also digital media
Media Center, new features, 32
media devices, synchronizing with, 31, 198-199
media files, playing (Windows Media Player 11), 200-201
Media Player 11. See Windows Media Player 11
Media Sharing (Windows Media Player 11), 199-200
members (collections), 323-324
memory
   adding, 391
   BIOS, optimizing startup, 389
   buffer overflow, 607
   cache memory, 6
   Command Prompt, modifying, 718
   flash memory, 250
   graphics memory, 9
   Memory Diagnostics Tool, 447-448
   requirements (Windows Vista), 6-7
   theoretical bandwidth, 7
   virtual
      optimizing, 398
      RADAR (Resource Exhaustion Detection and Resolution), 446-447
Memory Diagnostics Tool, 447-448
Memory Diagnostics Tool icon (Control Panel), 278
memory dumps, 442
Memory Leak Diagnosis, 21
menu accelerator keys. See accelerator keys
menus
   classic menus, viewing/hiding, 71
   converting
      Control Panel icons to, 285-286
      links (Start menu) to, 358-359
   timeouts, optimizing, 390
Message dialog box, 334
messages
   CDOs, 592
   email
      customizing columns, 524
      filtering, 537-538, 540
      incoming, 520-521
      managing, 535-537
      processing, 522-523
      reading, 525-527
      searching, 540-542
      sending, 527-534
   newsgroups
      customizing, 559-562
      decoding, 555
      downloading, 553-554
      filtering, 557
      managing, 555
      posting, 556-557
      rating posts, 558-559
      replying to, 555-556
      ROT13, unscrambling, 555
metadata
   applying to images (Windows Photo Gallery), 189-190
   defined, 65
   editing, 83-84
   filtering files, 92-93
grouping files, 91-92
overview, 82-85
stacking files, 92
support for, 18-19
Windows Media Player 11, 195, 197

methods
AddWindowsPrinterConnection, 344
CreateObject, 326
CreateShortcut, 337
GetObject, 328-329
InsertBefore, 327
Navigate, 346
objects, 321-322
Quit, 325
RegDelete, 341
RegRead, 339
RegWrite, 340
RemovePrinterConnection, 344
Save, 327

Microsoft codenames, 2
Microsoft Help Groups (Windows Mail), 29
Microsoft Knowledge Base website, 451
Microsoft Product Support Services website, 450
Microsoft Security website, 451
Microsoft TechNet website, 451
missing parameters, searching, 742-743
mobile computers. See notebook computers
Mobile PC Control Panel, 248
Mobility Center. See Windows Mobility Center
modems, installing, 462
modes
Batch, overview of, 725
Protected, 576-577
modifying. See also customizing; editing
contacts, 216-219
cover pages, 239
Disk Defragmenter scheduling, 419
folders, Command Prompt, 700
fonts, Command Prompt, 713
hard drives, offline files, 653-654
home pages, 505-506
listening ports, 673
memory, properties, 718
page files, 399-401
security zones, 576
spyware scanning, 572-573
views, Windows Calendar, 221-222
WshShell objects, 336-339

monitoring
battery life (notebook computers), 251-252
Event Viewer, 430, 432, 434
firewalls, 569
page files, 399
performance, 380
with Reliability and Performance Monitor, 385-389, 405
with Task Manager, 382-385
viewing performance rating, 380-382

monitors
attaching second monitor, 370
external monitors, attaching to notebook computers, 256
Moore's Law, 5
mouse, keyboard shortcuts, 685
Mouse icon (Control Panel), 280
Move File dialog box, 76
moving
e-mail messages, 523
files
resolving name conflicts, 76-77
Send To command, 78-79
insertion points (keyboard shortcuts), 685
user folders, 101-102

How can we make this index more useful? Email us at indexes@samspublishing.com
MP3 format, 205
msconfig -2, 390
multimedia. See digital media
multiple calendars, configuring, 229. See also calendars
multiple commands, DOSKEY, 702
multiple computers, checking email accounts, 519
multiple file types, associating with applications, 115
multiple home pages (Internet Explorer), 28
multiple time zones, viewing on taskbar, 365-367
music (Windows Media Player 11)
    copying to/from CDs, 204-208
    protecting, 206
My Business Card command, 219

N
name conflicts, resolving when copying files, 76-77
names
    arguments, 322
    DNS, 566
    IDNs, 579, 581-583
    long filenames, 699
    newsgroups, 546
    UNC, 635
NAP (Network Access Protection), 23
Native Command Queuing (NCQ), 8
Navigate method, 346
navigating
    dates, 220
    folders, 67-69
    Internet customizing security, 589-590
    Internet Explorer
    accessing Address bars, 483-484
    adding search engines, 490-491
    adding/removing zone sites, 575
    applying add-ons, 577
    applying tabs, 485-488
    blocking pop-ups, 587-588
    caching, 504-505
    configuring search engines, 491-493
    configuring security, 573-574
    creating URL shortcuts, 484-485
    customizing, 502
    deleting browser history, 584-585
    encoding addresses, 581-583
    formatting Favorites folders, 493-497
    Links bars, 502-503
    managing add-ons, 583
    modifying home pages, 505-506
    modifying zone sites, 576
    optimizing, 480-482
    optimizing privacy, 585-587
    phishing filters, 578-580
    reading RSS feeds, 497-499, 501
    reducing privileges, 576-577
    searching, 490
    setting options, 505, 507, 509-514
    subscribing RSS feeds, 499
    updating RSS feeds, 501-502
    viewing History lists, 488, 490, 506
    web page addresses, 480
    newsgroups, 546
    web pages, 346
    Windows Media Player 11, 193-194
Navigation pane, 73
NCQ (Native Command Queuing), 8
NET LOCALGROUP command, 166
NET USER command, 165-166
NetMeeting. See Windows Meeting Space
Network Access Protection (NAP), 23
Network and Sharing Center, 618, 620
customizing networks, 621-622
enabling/disabling network discovery, 622-623
new features, 35
viewing network map, 623
Network and Sharing Center icon (Control Panel), 280, 285
Network Configuration Operators group, 150
Network Connections dialog box, 625-626
Network Diagnostics, 21
network discovery
defined, 621
enabling/disabling, 622-623
Network icon, views of, 618
Network Map
new features, 35-36
viewing, 623
peer-to-peer networks
ad hoc wireless network connections, 617-618
changing computer and workgroup names, 611-612
nonbroadcasting wireless network connections, 615-616
setup, 609-611
wireless network connections, 612-615
printers, mapping, 344
remote connections, 663, 665-672
configuring, 664-665
disconnecting, 672
Internet, 673-676
VPNs, 677-681
Sync Center, 647
enabling offline files, 647-650
modifying disk space, 653-654
synchronizing offline files, 650-653
troubleshooting, IP address conflicts, 611
types of, 621
Windows Meeting Space, 654
closing, 661
inviting, 657-658
joining, 656
managing, 660
sharing handouts, 658
signing in, 655
starting, 655-656, 658-659
wireless networks
managing, 624-625
security settings, 625
New Call dialog box, 216
New Connection Security Rule Wizard, 569

How can we make this index more useful? Email us at indexes@samspublishing.com
new features of Windows Vista

- burning DVDs, 33
- Control Panel, 24, 26-27
- Desktop Window Manager (DWM), 23
- Game Explorer, 34-35
- graphics, 23
- interface features, 14-18
- Internet Explorer 7, 27-29
- Media Center, 32
- metadata support, 18-19
- Mobility Center, 35
- Network and Sharing Center, 35
- Network Map, 35-36
- per-application volume control, 33
- performance, 19-20
- security, 21-23
- Sound Recorder, 34
- stability, 20-21
- Transactional NTFS (TxF), 23
- Welcome Center, 24-25
- Windows Backup, 34
- Windows Calendar, 29-30
- Windows Easy Transfer, 34
- Windows Mail, 29
- Windows Media Player 11, 30, 32
- Windows Meeting Space, 36
- Windows Photo Gallery, 33
- Windows Presentation Foundation (WPF), 23
- XML Paper Specification (XPS), 24

New Inbound Rule Wizard, 569
New menu, customizing, 117-118
New Message window, 218
New News Rule dialog box, 557
news servers, 545
newsgroups, 451

Microsoft Help Groups (Windows Mail), new features, 29

Usenet
- articles/threads, 546
- configuring accounts, 549-550
- customizing, 559-562
- downloading messages, 553-554
- etiquette rules, 547-548
- filtering messages, 557
- managing messages, 555
- navigating, 546
- overview of, 544-545
- posting messages, 556-557
- rating posts, 558-559
- replying to messages, 555-556
- subscribing, 550-552

newsreaders, 544
NNTP servers, 545

nonbroadcasting wireless networks, connecting to, 615-616

notebook computers, 247. See also Tablet PCs
- external monitors, attaching, 256
- Mobile PC Control Panel, 248
- power plans, 250
  - configuring power buttons, 255-256
  - creating custom, 253-255
  - list of, 252-253
  - monitoring battery life, 251-252
- presentation settings, 257-259
- Windows Mobility Center, 35, 249-250
- Windows SideShow, 259-260

notification area (taskbar), hiding/viewing icons, 365

NTFS (NT file system)
- partitions, 143
  - converting, 396-397
- cycles, 409
- security, permissions, 645

NX Bit, 607
objects
CDOs, 592
  exposing, 329-330
properties (Internet Explorer), 347
scripts, programming, 319-324
variables, assigning, 322-323
WScript, 324
  automating, 325-331
WshNetwork, programming, 343-345
WshShell, programming, 331-333, 335-341, 343
offline files, 647
  enabling, 647-650
  modifying disk space, 653-654
  synchronizing, 650-653
Offline Files icon (Control Panel), 280
offline newsgroups, marking messages for downloading, 553
online resources for troubleshooting, 450-451
Open dialog box, file maintenance, 82
Open With command, launching applications, 133
Open With dialog box, 113-115
  customizing, 118
    adding applications to, 121
    deleting applications from, 121
    deleting applications from file types menu, 120
    disabling Always Open With check box, 122
    opening documents with unassociated applications, 119-120
    Registry settings, 120
opening. See also executing; launching; running
  command prompt
    as administrator, 128
    in current folder, 111-112
CPL files, 282-284
  shortcuts, Favorites folders, 494-495
System Properties dialog box, 43
web pages
  accessing Address bars, 483-484
  applying tabs, 485-488
  creating URL shortcuts, 484-485
  optimizing navigating, 480-482
  viewing History lists, 488, 490, 506
operating system modes.
  See compatibility layers
operators
  input redirection, 748-749
  pipe, 749-750
  redirection, 747
optimizing. See also performance
  applications, 391-392
    adding memory, 391
    configuring priority, 393-394
    installing fast hard drives, 391
    starting, 392
    upgrading device drivers, 392
  command line, 732-734
  Device Manager, 457-458
    hardware, 465-468, 470
    installing devices, 458
    legacy devices, 462-463
    Plug and Play devices, 458-460, 462
    security policies, 470-471
    signing options, 463-465
    troubleshooting, 472-477
  DOSKEY, 701-707
  hard disks, defragmenting, 417-419
  hard drives, 394
    closing Indexers, 395
    configuring Last Access Time, 397
    converting partitions to NTFS, 396-397
disabling compression/encryption, 395
enabling write caching, 395-396
implementing maintenance, 394
turning off 8.3 filename creation, 397
viewing specifications, 394

Internet Explorer
accessing Address bars, 483-484
applying tabs, 485-488
creating URL shortcuts, 484-485
navigating, 480-482
viewing History lists, 488, 490, 506
keyboard shortcuts, 683-689
maintenance, scheduling, 434-435
privacy, 585-587, 599-600
stability improvements, 403
I/O cancellation, 404
Reliability Monitor, 405
service recovery, 406
Startup Repair Tool, 407

startup, 389
closing splash screens, 390
configuring prefetchers, 391
logging on automatically, 391
reducing BIOS checks, 389
speeding up, 390
upgrading device drivers, 390
updates, 428-429
Checking for security vulnerabilities, 430-431
virtual memory, 398
customizing page files, 398-399
modifying page files, 399-401
monitoring page files, 399
saving page files, 398
splitting page files, 398

options
Command Prompt, 711-712, 714, 716, 718-721, 723-724
email
configuring reading, 525-527
sending, 531-534
environment variables, 738-739
faxes
receiving, 242-244
sending, 236-238
Internet Explorer, configuring, 505, 507, 509-514
maintenance, scheduling, 434-435
newsgroups, 559-562
passwords, 168-169
playback options (Windows Media Player 11), 202-203
pointer options (Tablet PCs), setting, 268-269
Recycle Bin, setting, 80-81
scripts, properties, 318-319
security
policies, 470-471
zones, 576
signing, 463-465
Tablet PCs, setting, 262-263
Windows Error Reporting service, 404
Windows Media Player, security, 605-606

Organize button (folders), 70
orientation. See screen orientation
output commands, redirecting, 747
outages, power. See troubleshooting
outbound rules, Windows Firewall, 569
Outlook Express. See Windows Mail
overflow buffers, 607
overriding automatic logon, 57
page files
  customizing, 398-399
  memory dumps and, 442
  modifying, 399-401
  monitoring, 399
  saving, 398
  splitting, 398

parameters
  batch files, 729-735
  Command Prompt, adding, 697-698
  searching, 742-743
  testing, 741-742

Parental Controls, 172-173
  activating, 173-175
  games example, 175-178
  new features, 23

Parental Controls icon (Control Panel), 280

Parkinson’s Law of Data, 5, 373

partition tables, 40

partitions
  Check Disk, running, 410-412
  clusters, 408-409
  FAT file system, 143
  files, deleting unnecessary, 415-417
  NTFS, 143
    converting, 396-397
    cycles, 409

password hints, 157, 170

password policies, 169-170

Password Reset Disks, 170-171

passwords
  forgotten passwords, recovering, 170-171
  options, 168-169
  password policies, 169-170
  for Sleep mode, disabling, 256
  strong passwords, creating, 167-168

pasting
  GUIDs (globally unique identifiers), 51
text (Command Prompt), 710-711

patches, Windows Mail, 536

PATH command, 726-727

paths
  application-specific paths
    creating, 140-141
    in Registry, 132
  defined, 132
  document-specific paths, creating, 141

PAUSE command, 729

payloads, 677

peer-to-peer networks
  ad hoc wireless network connections, 617-618
  computer and workgroup names, changing, 611-612
  nonbroadcasting wireless network connections, 615-616
  setup, 609-611
  wireless network connections, 612-615

Pen and Input Devices icon (Control Panel), 280

pen flicks (Tablet PCs), 266-268

People Near Me icon (Control Panel), 280

per-application volume control, 33, 184-186

perceive types, defined, 120

performance. See also optimizing
  applications
    adding memory, 391
    configuring priority, 393-394
    installing fast hard drives, 391
    optimizing, 391-392
    starting, 392
    upgrading device drivers, 392
  hard drives
    closing Indexers, 395
    configuring Last Access Time, 397

How can we make this index more useful? Email us at indexes@samspublishing.com
converting partitions to NTFS, 396-397
disabling compression/encryption, 395
enabling write caching, 395-396
implementing maintenance, 394
optimizing, 394
turning off 8.3 filename creation, 397
viewing specifications, 394
monitoring, 380
with Reliability and Performance Monitor, 385-389
with Task Manager, 382-385
viewing performance rating, 380-382

new features, 19-20
startup
  closing splash screens, 390
  configuring prefetchers, 391
  logging on automatically, 391
  optimizing, 389
  reducing BIOS checks, 389
  speeding up, 390
  upgrading device drivers, 390

virtual memory
  customizing page files, 398-399
  modifying page files, 399-401
  monitoring page files, 399
  optimizing, 398
  saving page files, 398
  splitting page files, 398

Windows Vista improvements, 374
  Restart Manager, 378, 380
  Sleep mode, 375-376
  startup speed, 374-375
  SuperFetch with ReadyBoost, 376-378

Performance Information and Tools icon (Control Panel), 280
Performance Log Users group, 150
Performance Monitor Users group, 150
performance rating, viewing, 380-382
Performance Rating tool, 381-382
permissions
  defined, 149
  Registry security, 301
  security, NTFS, 645
Personalization icon (Control Panel), 280, 284
personalizing handwriting recognition (Tablet PCs), 268-271
phishing, filtering, 578-580, 598
Phishing Filter, 22
Phone and Modem Options icon (Control Panel), 280
photos. See images
pictures. See images
PIFs (program information files), 698
pinned programs, defined, 355
pinned programs list (Start menu), customizing, 357-358
piping commands, 749-750
playlists, 207
playback options (Windows Media Player 11), 202-203
playing
  DVDs, new features, 32
  media files (Windows Media Player 11), 200-201
policies
  account lockout policies, setting, 164
  group policies, User Account Control, 155-156
  password policies, 169-170
  security policies
    devices, 470-471
    setting, 161-162
  user rights policies, setting, 163-164
pop-ups, blocking, 587-588
Popup dialog box, 333
portable computers. See notebook computers
ports
  forwarding, 676
  listening, modifying, 673
  SMTP, specifying, 519
POST (Power-On Self Test), 40, 389
posting messages
  newsgroups, 545, 556-557
  rating, 558-559
power buttons (notebook computers), configuring, 255-256
Power Options icon (Control Panel), 280, 285
power plans, notebook computers, 250
  configuring power buttons, 255-256
  creating custom power plans, 253-255
  list of power plans, 252-253
  monitoring battery life, 251-252
power surges, lost clusters, 409.
  See also troubleshooting
Power Users group, 151
Power-On Self Test (POST), 40, 389
pre-installation checklist, installing applications, 124-130
preferences
  Command Prompt, 711-712, 714, 716, 718-721, 723-724
  email
    configuring reading, 525-527
    sending, 531-534
  environment variables, 738-739
  faxes
    receiving, 242-244
    sending, 236-238
  Internet Explorer, configuring, 505, 507, 509-514
  maintenance, scheduling, 434-435
  newsgroups, 559-562
  scripts, properties, 318-319
  security
    policies, 470-471
    zones, 576
    Windows Error Reporting service, 404
    Windows Media Player, security, 605-606
prefetchers
  configuring, 391
  defined, 376
presentation settings
  notebook computers, 257-259
  Windows Mobility Center, 250
preventing IDN spoofing, 581-583.
  See also disabling
Preview pane, 72
previous versions of volumes/folders/files, reverting to, 94-95
primary names, defined, 132
Print Management icon (Control Panel), 278
printers
  installing, 462
  networks, mapping, 344
  sharing, 639-643, 645-647
Printers icon (Control Panel), 280
printing
  messages, 523
  networks, 639
priority
  configuring, 393-394
  of networks, reordering, 624
privacy
  optimizing, 585-587, 599-600
  Windows Media Player, 605-606
private networks, defined, 621
privileges, 151-152
  defined, 149
  elevating, 153-154
file and Registry virtualization, 154
reducing, 576-577
User Account Control, new features, 21

Problem Reports and Solutions, 448-450
Problem Reports and Solutions icon (Control Panel), 280
problems. See also troubleshooting
causes of, 123, 437
determining, 438-444
recovery
last known good configuration, booting from, 452
System Recovery Options, 454-455
System Restore, 452-454
processing email messages, 522-523
customizing columns, 524
filtering, 537-538, 540
managing, 535-537
reading, 525-527
searching, 540-542
sending, 527-534
processor requirements (Windows Vista), 5-6
processor scheduling, 392
profiles. See user profiles
PROGMAN.INI file, 292
program information files. See PIFs
program settings (Registry), 131
programming
FileSystemObject, 330-331
objects, scripts, 319-324
WshNetwork objects, 343-345
WshShell objects, 331-333, 335-341, 343
Programs and Features icon (Control Panel), 281
programs. See applications
properties. See also metadata
audio devices, 187-188
Command Prompt, customizing, 717
Device Manager, viewing, 467
memory, modifying, 718
objects, 320
Internet Explorer, 347
screens, 720
scripts, customizing, 318-319
WshNetwork objects, 343
Protected mode (Internet Explorer), 22, 40, 576-577
protecting
email
spam, 594-598
viruses, 591, 593-594
music (Windows Media Player 11), 206
protocols
DHCP, 610
FTP, 566
WEP, 614
WPA, 614
Public folder, 171-172
public keys, sending/receiving secure email, 602
public networks, defined, 621
public news servers, 545
publishing calendars. See calendars
Punycode values, 582

Q

QueryString, 492
Quick Launch toolbar, 362
Quick Scan, Windows Defender, 571
Quick Tabs. See tabs
Quit method, 325
quitting applications (Task Manager), 443
QWORD values (in Registry), editing, 305-306
RADAR (Resource Exhaustion Detection and Resolution), 21, 446-447
RAM, adding, 391
rating posts, newsgroups, 558-559
ratings, games, 177-178
read receipts, blocking, 599
reading
  email, 525-527
    requesting read receipts, 529
  newsgroups, customizing, 559
  RSS feeds, 497-499, 501
Readme.txt files, pre-installation checklist, 125
ReadyBoost, 376-378
Real mode, 40
Real Simple Syndication. See RSS feeds
rebooting
  causes of problems, determining, 440-442
  for updates with Restart Manager, 378, 380
recalling command lines, 701
receipts (read), blocking, 599
receiving
  email, security, 600-604
  faxes, 242-244
recent programs list (Start menu), clearing, 354
recovering
  deleted files, 81
  forgotten passwords, 170-171
recovery
  last known good configuration, booting from, 452
  services, 406
  System Recovery Options, 454-455
  System Restore, 452-454
Recurrence dialog box, 225
recurring appointments, creating.  
  See Windows Calendar

Recycle Bin, 79-80
  options, setting, 80-81
  recovering deleted files, 81
redirecting Windows Vista, 747-749
reducing
  BIOS checks, 389
  privileges, 576-577
references
  WshNetwork objects, 343
  WshShell objects, 332
.reg file
  comments in, 308
  creating, 307-309
  editing, 307
  exporting Registry as, 301-302
  exporting Registry keys to, 302
  importing, 303
RegDelete method, 341
Regional and Language Options icon (Control Panel), 281
registering file extensions, 115-116
Registry
  AutoRun settings, 693
  causes of problems, determining, 440
  entries
    changing values of, 303, 305-307, 309
    finding, 310-311
  file and Registry virtualization, 154
  file types and, 106-108
    ShellNew subkey, 117-118
  finding changes to, 302
  history of, 293
  hives, 299-300
  installing applications, 131-132
  keys
    creating, 310
    deleting, 310

How can we make this index more useful? Email us at indexes@samspublishing.com
renaming, 309
saving as favorites, 304
launching applications at startup, 135-136
legacy keys, defined, 136
Open With dialog box settings, 120
purpose of, 291-292
Registry Editor, 294
disabling access to, 300
finding entries, 310-311
Keys pane, 295
Settings pane, 296
root keys, 296
HKEY_CLASSES_ROOT, 296-297
HKEY_CURRENT_CONFIG, 298
HKEY_CURRENT_USER, 297-298
HKEY_LOCAL_MACHINE, 298
HKEY_USERS, 298
security of, 300
backing up, 301
exporting keys, 301-303
permissions, 301
System Restore, 301
WshShell object, 339-341
Registry Editor, 294
disabling access to, 300
finding entries, 310-311
Keys pane, 295
Settings pane, 296
RegRead method, 339
RegWrite method, 340
reinstalling drivers, 474
Reliability and Performance Monitor, 385-389
Reliability and Performance Monitor icon (Control Panel), 278
reliability events, 20
Reliability Monitor, 20, 405
REM command, 728
reminders (Windows Calendar), adding, 225
remote connections
Remote Desktop, 663, 665-672
configuring, 664-665
disconnecting, 672
Internet, 673-676
VPNs, 677-681
Remote Desktop Users dialog box, 664
Remote Desktop Users group, 151
removable drives, mapping, 637
RemovePrinterConnection method, 344
renaming
computers and workgroups, 611-612
entries in Windows Boot Manager, 51
files with hidden file extensions, 100
network connections, 626
Registry keys, 118, 309
Start menu items, 358
reordering
entries in Windows Boot Manager, 51
network priorities, 624
Repair tool, troubleshooting networks, 633-634
Replicator group, 151
replying
applying different addresses, 518
email messages, 523
newsgroups messages, 555-556
reports, data collectors sets, 389
requesting read receipts, email, 529
requirements. See system requirements
resizing toolbars, 362
resolution. See screen resolution
Resolve Conflict dialog box, 653
resolving name conflicts when copying files, 76-77
Resource Exhaustion Detection and Resolution (RADAR), 446-447
resources
- conflicts, troubleshooting, 476-477

networks
- accessing, 634-635
- adding, 632
- creating locations, 638
- mapping folders, 636-638
- printing, 639
- sharing accessing, 639-643, 645-647
- viewing, 632

Restart Manager, 20, 378, 380

restore points. See system restore points

restoring with System Restore, 421-423

restricting games by ratings, 177-178

retrieving email, 520-521
- processing, 522-523

returning values (properties), 321

reversible encryption, 170

reverting to previous versions of volumes/ folders/files, 94-95

rights, defined, 149

ripping
- audio CDs (Windows Media Player 11), 204-207
- files, new features, 31

roaming profiles, defined, 151

rogue services, preventing, 606

rolling back device drivers, 470, 474

rolling back transactions, 93

root keys (Registry), 295-296
- HKEY_CLASSES_ROOT, 296-297
- HKEY_CURRENT_CONFIG, 298
- HKEY_CURRENT_USER, 297-298
- HKEY_LOCAL_MACHINE, 298
- HKEY_USERS, 298

ROT13 messages, unscrambling, 555

rotating images (Windows Photo Gallery), 191

RSS (Real Simple Syndication) feeds, 28, 497-499, 501
- new features, 28
- subscribing, 499
- updating, 501-502

rules
- connections, security, 569
- email, filtering, 538
- etiquette (newsgroups), 547-548
- Windows Firewall, 568

Run command, adding to Start menu, 133

Run dialog box
- .exe file extension in, 141
- launching applications, 133-134

RUNAS command, 139

running. See also executing; launching; opening
applications (WshShell object), 336
Check Disk, 410-412
Disk Defragmenter, 417-419
logon scripts, 138
scripts, 315
setup.exe, 130
System Restore, 421-423

S

Safe mode, 45, 53
- troubleshooting, 59-60
- when to use, 58

Safe Senders, configuring, 596

Save As dialog box, file maintenance, 82

Save method, 327

saving
- directory listings as text files, 127-129
- email attachments, 522
- email messages, 523
page files, 398
Registry keys as favorites, 304
searches, 90
scanners, installing, 462
Scanners and Cameras icon (Control Panel), 281
scanning spyware, 571-572
configuring Windows Defender, 572-573
scheduling
data collector sets, 389
Disk Defragmenter, modifying, 419
maintenance, 434-435
offline file synchronization, 650-652
processor, 392
RSS feeds, updating, 501-502
spyware scans, 572
Windows Calendar, 219-220
configuring multiple calendars, 229
importing files, 230
modifying views, 221-222
navigating dates, 220
sharing, 231-233
tracking appointments, 222-228
scratch-out gestures (Tablet PCs), 265
screen orientation of Tablet PCs, changing, 262
screen orientation setting (Windows Mobility Center), 250
screen resolution
changing, 370-371
defined, 369-370
screens, properties, 720
scripts
compatibility layers, 145, 147-148
launching at startup, 134-139
text, viewing, 325
WSH, 313, 315
applying CScript, 317
applying WScript, 316-317
customizing properties, 318-319
executing, 315
Internet Explorer, 345-346, 348-349
programming objects, 319-324
running, 315
WScript objects, 324-331
WshNetwork objects, 343-345
WshShell objects, 331-333, 335-341, 343
ScriptURL, 492
SDMI (Secure Digital Music Initiative), 206
Search box in Windows Mail, new features, 29
searches
advanced searches, 88-90
e-mail, 540-542
images (Windows Photo Gallery), 191
Instant Search box, 17, 31, 69-70, 86-88
Internet Explorer, 490
adding search engines, 490-491
configuring search engines, 491-493
metadata, 18
missing parameters, 742-743
saving, 90
Windows Media Player 11, 197
Windows Search Engine, 85-86
second monitor, attaching, 370
secondary displays (Windows SideShow), 259-260
sectors, clusters, 408-409
Secure Digital Music Initiative (SDMI), 206
Secure Startup, new features, 22
security
ASLR, 608
Control Panel, configuring, 564
e-mail, 590
encrypting, 529
optimizing privacy, 599-600
phishing filters, 598
customizing, 531-534
security, 600-604
faxes, 236-238

servers
news, 545

VPNs. See VPNs

service recovery policy, new features, 20

Service Set Identifier (SSID), disabling broadcasting, 615

services
defined, 62
recovery, 406
Windows Error Reporting, 404
Windows Service Hardening, 606

Services icon (Control Panel), 278

sessions
Command Prompt, 691
adding parameters, 697-698
CMD.EXE switches, 692, 694-695
configuring, 699-707
executing commands, 696
Windows Meeting Space. See Windows Meeting Space

Set Associations dialog box, 114
Set Default Programs dialog box, 115
Set Program Access and Computer Defaults dialog box, 356
Settings pane (Registry Editor), 296
setup.exe, running, 130

shadow copies, defined, 94

sharing
calendars, 231-233
Command Prompt, applications, 709-711
computers, 179
files, Public folder, 171-172
handouts, Windows Meeting Space, 658
Media Sharing (Windows Media Player 11), 199-200
resources, networks, 634-635, 639-643, 645-647
shortcuts, Favorites folders, 496-497
Sharing Wizard, deactivating, 641-642
sharpening text with ClearType, 370
ShellNew subkey (Registry), 117-118
SHIFT command, 734-735
shortcut keys, Windows Media Player 11, 201

shortcuts
customizing in Start Menu folder, 360
defined, 112
dragging/dropping onto Start button, 360
e-mail, creating, 530
Favorites folders
adding, 494
managing, 495-496
opening, 494-495
sharing, 496-497
keyboards, 683-689
URLs, creating, 484-485
WshShell object, 336-339

shutdown. See quitting
shutting down scripts, 325
side-by-side compliant DLLs, defined, 379
Sidebar. See Windows Sidebar
SideShow. See Windows SideShow
SIDs, determining, 297
Signature Verification Tool, 474
signatures, e-mail
adding, 529
creating, 530-534
signing options, 463-465
Size on Disk value, Size value versus, 75
Size value, Size on Disk value versus, 75
sizes, clusters, 408-409
sizing. See resizing
SKU (stock keeping unit), 12
Sleep mode, 375-376
    new features, 19
    password requirements, disabling, 256
SMART (Self-Monitoring, Analysis, and Reporting Technology), 446
SMTP
    authentication, enabling, 518
    ports, specifying, 519
Snipping Tool (Tablet PCs), 272
social bookmarking, defined, 83
social software, defined, 83
Software Explorer, disabling application launches at startup, 141-142
software. See applications
SORT command, 749
sound, incoming email messages, 520.
    See also audio
Sound icon (Control Panel), 281, 285
sound quality, compression versus, 206
Sound Recorder, 34, 186-187
spam email, 594-598
spam filters. See Junk Mail Filter
speakers, volume control, 186
specifications, viewing hard drives, 394
Speech Recognition Options icon (Control Panel), 281
speed at startup, optimizing, 390.
    See also performance
splash screens, closing, 390
splitting page files, 398
spoofing IDN, preventing, 581-583.
    See also phishing; security
spyware, 22, 570-571. See also security
    Protected mode, 576-577
    scanning, 571-572
    configuring Windows Defender, 572-573
SRT (Startup Repair Tool), 21, 60, 407, 455
SSID (Service Set Identifier), disabling broadcasting, 615
stability improvements, 403
    I/O cancellation, 404
    new features, 20-21
    Reliability Monitor, 405
    service recovery, 406
    Startup Repair Tool, 407
stacking
    files, 18, 92
    media, new features, 31
    in Windows Media Player 11, 194, 196
Standard users group, 152
Standby mode, 375
Start button
    dragging/dropping shortcuts on, 360
    new features, 14
Start menu
    adding Run command to, 133
    Control Panel icons, converting to menu, 285-286
    customizing, 352
        All Programs menu icons, 359, 361
        clearing recent programs list, 354
        converting links to menus, 358-359
        default programs, setting, 356-357
        favorite programs area, 352-354
        with group policies, 368-369
        Internet and E-mail icons, 355-356
        pinned programs list, 357-358
    Instant Search box, 86-88
    launching applications, 133
    new features, 14
    renaming Start menu items, 358
Start Menu folder, customizing shortcuts, 360
starting. See also executing; launching; running; startup process
    applications
        optimizing, 392
        WshShell objects, 336

How can we make this index more useful? Email us at indexes@samspublishing.com
Command Prompt, 165, 691
applications, 707, 709
CMD.EXE switches, 692, 694-695
Windows Fax and Scan, 234-235
Windows Meeting Space, 655-656, 658-659
Startup and Recovery dialog box, 441
customizing BCD options, 43-44
Startup folder, 134-135
startup process. See also logon process
  BIOS settings, adjusting, 454
  from CD, 60
customizing
    with Advanced Boot Options menu, 52-54
    with BCD (Boot Configuration Data), 41-50, 52
disabling application launch at,
  135, 141-142
e-mail, sending/receiving, 521
launching applications and scripts, 134-139
new features, 19
Secure Startup, new features, 22
speed of, 374-375
steps in, 39-41
system failures during, 441
troubleshooting
  advanced startup options, when to use, 58-59
  with CompletePC Restore, 64
  Safe mode won’t start, 59-60
  with System Configuration Utility, 61-64
  with System Recovery Options, 60-61
  with System Restore, 64
Startup Repair Tool (SRT), 21, 60, 407, 455
stationary
  applying to e-mail, 528
  saving e-mail messages as, 523
stop errors. See system failures
stopping. See quitting
storage requirements (Windows Vista), 7-8
storing Registry settings, 340
string values in Registry, editing, 305
strings, QueryString, 492
strong passwords, creating, 167-168
subscribing
  calendars, 232
  newsgroups, 545, 550-552
  RSS feeds, 499
SuperFetch, 376-378
  configuring, 391
  new features, 19
suspending. See overriding
switches
  for BCDEDIT, 48
  CMD.EXE, 692, 694-695
  Command Prompt, adding, 697-698
Sync Center, 647
  offline files
    enabling, 647-650
    modifying disk space, 653-654
    synchronizing, 650-653
Sync Center icon (Control Panel), 281
sync center setting (Windows Mobility Center), 250
synchronizing
  with media devices, 31
  messages, 553
  offline files, 650-652
    troubleshooting, 652-653
  Windows Media Player 11 with media devices, 198-199
System Configuration icon (Control Panel), 278
System Configuration Utility
customizing BCD options, 44-47
troubleshooting startup process, 61-64
system failures
  configuring handling of, 440-442
  during startup, 441
  lost clusters, 409
System icon (Control Panel), 281, 285
system images, creating, 427
System Information utility, determining causes of problems, 439-440
System Properties dialog box, 43, 612, 664
System Recovery Options, 454-455
  troubleshooting startup process, 60-61
system requirements (Windows Vista), 5
  graphics, 8-9
  hardware, 9-10
  memory, 6-7
  processors, 5-6
  storage, 7-8
System Restore, 60, 452-455
  configuring, 421-423
  Registry security, 301
  troubleshooting startup process, 64
system restore points, setting (pre-installation checklist), 125
System Stability Chart, 405
system state, defined, 301
SYSTEM.INI file, 292

T

tab numbering in dialog boxes, 283
tabbed browsing (Internet Explorer), new features, 27
Tablet PC Input Panel toolbar, 361
Tablet PC Settings icon (Control Panel), 281
Tablet PCs, 260, 262. See also notebook computers
  handwriting recognition, personalizing, 268-271
  Input Panel, 263-265
  options, setting, 262-263
  pen flicks, 266-268
  pointer options, setting, 268-269
  screen orientation, changing, 262
  Snipping Tool, 272
tabs, applying, 485-488
tags
  applying to images (Windows Photo Gallery), 189-190
  defined, 83
  job, 317
Windows Media Player 11, 195, 197
Task Manager
  force quitting applications, 443
  performance monitoring, 382-385
  priority, configuring, 393-394
Task pane, 70
Task Scheduler
  launching applications, 134
  launching applications at startup, 138-139
Task Scheduler icon (Control Panel), 278
taskbar
  customizing
    creating toolbars, 362
    customizing toolbars, 362
    with group policies, 368-369
    grouping taskbar buttons, 367-368
    setting options, 363-365
    viewing multiple time zones, 365-367
    viewing toolbars, 361-362
  unlocking, 362, 367
Taskbar and Start Menu icon (Control Panel), 281, 285
Taskbar and Start Menu Properties dialog box, 363
taskbar thumbnails, new features, 16
tasks
  deleting, 229
  maintenance, scheduling, 434-435
  tracking, 222-228
  Windows Calendar. See Windows Calendar
technical support. See online resources
testing
  parameters, 741-742
  POST, 389
text
  Command Prompt
    copying, 709-710
    pasting, 710-711
  fields, adding, 241
  keyboard shortcuts, 685
  REM command, 728-729
  scripts
    applying CScript, 317
    applying WScript, 316-317
    customizing properties, 318-319
    executing, 315
    running, 315
    viewing, 325
    sharpening with ClearType, 370
text files. See also documents
  comparing, 128-129
  saving directory listings as, 127-129
Text to Speech icon (Control Panel), 281
theoretical bandwidth of memory, 7
threads (newsgroups), 545-546
  managing, 555
thumbnails
  taskbar thumbnails, 16
  window thumbnails, 15
time slices, 723-724
time zones, viewing multiple on taskbar, 365-367
timeouts, optimizing, 390
toolbars
  Cover Page Editor Drawing, 241
  resizing, 362
  taskbar toolbars
    creating, 362
    customizing, 362
    viewing, 361-362
tools
  contact management, 209, 211-212
  communicating with, 216
  creating groups, 214-215
  importing, 212-214
  modifying, 216-219
  Device Manager, 457-458
  hardware, 465-468, 470
  installing devices, 458
  legacy devices, 462-463
  Plug and Play devices, 458-460, 462
  security policies, 470-471
  signing options, 463-465
  troubleshooting, 472-477
diagnostic tools, 445
  disk diagnostics, 446
  Memory Diagnostics Tool, 447-448
  Problem Reports and Solutions, 448-450
  RADAR (Resource Exhaustion Detection and Resolution), 446-447
  Disk Cleanup, 415
  Disk Defragmenter, 417-419
  Event Viewer, 430, 432, 434
  Microsoft Backup, 424
    automating, 425, 427
    creating system images, 427
  Repair, troubleshooting networks, 633-634
  Signature Verification, 474
  Startup Repair Tool, 407
in System Recovery Options, 455
System Restore, 421-423

tracking
appointments, 222-228.
See also Windows Calendar
environment variables, 341, 343
Favorites folders, 493
adding shortcuts, 494
managing shortcuts, 495-496
opening shortcuts, 494-495
sharing shortcuts, 496-497
hard disk free space, 412-414
Reliability Monitor, 405

Transaction Monitor (TxF), 23, 93-94
transactions, defined, 93

Transmitting Subscriber Identification. See TSID

Trojan horse, 567

Troubleshooters, defined, 445
troubleshooting. See also diagnostics; problems
backups, 424
automating, 425, 427
creating system images, 427
booting from DVD, 454
causes of problems, 123, 437
determining, 438-444
Device Manager, 472-477
Disk Defragmenter, 417-419
Event Viewer, 430, 432, 434
hard disks
checking for errors, 407-409
running Check Disk, 410-412
IP address conflicts, 611
networks, 633-634
newsgroups, 560
offline file synchronization, 652
with online resources, 450-451
scheduling maintenance, 434-435
security
configuring Windows Defender, 572-573
scanning spyware, 571-572
spyware, 570-571
stability improvements, 403
I/O cancellation, 404
Reliability Monitor, 405
service recovery, 406
Startup Repair Tool, 407
startup process
advanced startup options, when to use, 58-59
with Complete PC Restore, 64
Safe mode won’t start, 59-60
with System Configuration Utility, 61-64
with System Recovery Options, 60-61
with System Restore, 64
System Restore, 421-423
tips for, 445
updates, 428-429
checking for security vulnerabilities, 430-431
with Windows Diagnostic Infrastructure, 445
disk diagnostics, 446
Memory Diagnostics Tool, 447-448
Problem Reports and Solutions, 448-450
RADAR (Resource Exhaustion Detection and Resolution), 446-447
XCOPY command, 745-746

Truscott, Tom, 544

TSID (Transmitting Subscriber Identification), 236
turning off. See disabling; hiding
turning on. See enabling; viewing

TxF (Transactional NTFS), 23, 93-94
UI (user interface). See interface

unassociated applications
defined, 119
opening documents with, 119-120

UNC (universal naming convention), 635

undoing commands, 81

Uniform Resource Locators. See URLs

uninstalling. See also installing
applications, 126-127, 415-417
devices, 470

universal naming convention. See UNC
unlocking taskbar, 362, 367
unnecessary files, deleting, 415-417

unscreaming ROT13 messages, 555

unsigned drivers, disabling, 474

unsubscribing from newsgroups, 552

updates, 428-429
causes of problems, determining, 444
drivers, 467-468, 470
rebooting with Restart Manager, 378, 380
RSS feeds, 501-502
security, checking for vulnerabilities, 430-431
Windows Mail, 536
Windows Media Player, disabling, 202

Upgrade Advisor tool, 144

upgrading
device drivers, 390, 392, 474
Windows Vista with Windows Anytime Upgrade feature, 12-14

URGE store (Windows Media Player 11), new features, 31

URLs (Uniform Resource Locators), 480
Favorites folders, 493
adding shortcuts, 494
managing shortcuts, 495-496
opening shortcuts, 494-495
sharing shortcuts, 496-497
phishing filters, 578-580, 598
ScriptURL, 492
shortcuts, creating, 484-485

Usenet
accounts, configuring, 549-550
articles/threads, 546
customizing, 559-562
etiquette rules, 547-548
messages
downloading, 553-554
filtering, 557
managing, 555
posting, 556-557
rating posts, 558-559
replying to, 555-556
navigating, 546
overview of, 544-545
subscribing, 550-552

User Account Control, 152
elevating privileges, 153-154
file and Registry virtualization, 154
group policies, 155-156
installing applications, 124
new features, 21

user accounts
account lockout policies, setting, 164
command prompt
NET LOCALGROUP command, 166
NET USER command, 165-166
starting session, 165
creating, 156
defined, 149
editing, 156-158
least privileged user, defined, 152
limited user accounts, defined, 152
Local Users and Groups snap-in, 160-161
passwords
  forgotten passwords, recovering, 170-171
  options, 168-169
  password policies, 169-170
  strong passwords, creating, 167-168
privileges, 151-152
  elevating, 153-154
file and Registry virtualization, 154
security groups, 149-151
security policies, setting, 161-162
User Accounts dialog box, 158-159
user rights policies, setting, 163-164
User Accounts dialog box, 158-159
User Accounts icon (Control Panel), 281
user folders, moving, 101-102
user interface (UI). See interface
user profiles, 151
user rights policies, setting, 163-164
user settings (Registry), 131
users
  adding with User Accounts dialog box, 158-159
  groups, adding, 161
Users group, 150

V

values
  properties
    configuring, 320
    returning, 321
Registry, WshShell objects, 339-341
variables
  delayed environment expansions, 738-739
  environment, WshShell objects, 341, 343
  objects, assigning, 322-323
VBScript
  objects, exposing, 329-330
  WSH, 313, 315
version numbers (Windows), 3
video. See digital media
view options (Windows Explorer), customizing, 98-101
viewing. See also displaying
classic menus, 71
Control Panel, 275-277
Device Manager, 465-466
devices, networks, 632
drive letters, 100
environment variables, 341, 343
Event Viewer, 430, 432, 434
file extensions, 100, 105
Folders list, 73
hard disk free space, 412-414
hard drive specifications, 394
hidden files, 99
History lists, 488, 490, 506
Internet Explorer. See Internet Explorer
multiple time zones on taskbar, 365-367
network map, 623
notification area (taskbar) icons, 365
performance rating, 380-382
Registry, WshShell objects, 339-341
shortcuts, 336-339
taskbar toolbars, 361-362
text, scripts, 325
user information (WshShell object), 332-333, 335
web pages, 346
workgroups, 635
views

Windows Calendar, modifying, 221-222
Windows Explorer, customizing, 95
Windows Media Player 11, 194, 196
Views button (folders), 70
virtual memory, RADAR (Resource Exhaustion Detection and Resolution), 446-447
virtual private networks. See VPNs
virus checks, pre-installation checklist, 125-126
viruses, email, 591, 593-594. See also security
Vista. See Windows Vista

volume

per-application volume control, 33, 184-186
reverting to previous versions, 94-95

volume setting (Windows Mobility Center), 249

VPNs (virtual private networks), 677-678, 681
clients, configuring, 679-680
gateways, configuring, 678-679

vulnerabilities (security), checking for, 430-431

W

WAV format, 205
WCF (Windows Communications Foundation), 4
WDDM (Windows Vista Display Driver Model), 9

web bugs, 599-600

web pages. See also Internet Explorer
addresses, 480
Favorites folders
adding shortcuts, 494
formatting, 493
managing shortcuts, 495-496
opening shortcuts, 494-495
sharing shortcuts, 496-497
navigating
accessing Address bars, 483-484
applying tabs, 485-488
creating URL shortcuts, 484-485
optimizing, 480-482
viewing History lists, 488, 490, 506
RSS feeds, 497-499, 501
subscribing, 499
updating, 501-502
searching, 490
adding search engines, 490-491
configuring search engines, 491-493

security
adding/removing zone sites, 575
applying add-ons, 577
blocking pop-ups, 587-588
configuring, 573-574
customizing, 589-590
deleting browser history, 584-585
encoding addresses, 581-583
managing add-ons, 583
modifying zone sites, 576
optimizing privacy, 585-587, 599-600
phishing filters, 578-580
reducing privileges, 576-577
viewing, 346

websites. See online resources

Welcome Center
disabling, 281
new features, 24-25

Welcome Center icon (Control Panel), 281

Welcome screen, including Administrator account in, 56

well-known identifiers, 50

WEP (Wired Equivalent Privacy), 614

Wi-Fi Protected Access (WPA), 614

WIN.INI file, 292

window thumbnails, new features, 15

windows. See folders
Windows

configuring, determining causes of problems, 440
history of releases, 1-2
version numbers, 3

Windows Anytime Upgrade feature, 12-14
Windows Backup, new features, 34

Windows Boot Manager (BOOTMGR), 40
displaying manually, 52

Windows Calendar
new features, 29-30
scheduling, 219-220
configuring multiple calendars, 229
importing files, 230
modifying views, 221-222
navigating dates, 220
sharing, 231-233
tracking appointments, 222-228

Windows CardSpace icon (Control Panel), 281

Windows Communications Foundation (WCF), 4

Windows Contacts, 209
communicating with, 216
groups, creating, 214-215
importing, 212-214
managing, 211-212
modifying, 216-219

Windows Defender
application launches at startup, disabling, 141-142
new features, 22
spyware, 570-571
configuring Windows Defender, 572-573
scanning, 571-572
in Windows XP 4

Windows Defender icon (Control Panel), 282

Windows Diagnostic Console, 21
defined, 445
disk diagnostics, 446

Memory Diagnostics Tool, 447-448
Problem Reports and Solutions, 448-450
RADAR (Resource Exhaustion Detection and Resolution), 446-447

Windows Easy Transfer, new features, 34

Windows Error Reporting service, 404

Windows Explorer, customizing. See also files; folders
changing views, 95
Details view, 96-97
full-screen mode, 97
moving user folders, 101-102
view options, list of, 98-101

Windows Fax and Scan. See also faxes; scanning
accounts, creating, 235
cover pages, formatting, 235, 239-241
starting, 234-235

Windows Firewall
configuring, 566-569
new features, 22

Windows Firewall icon (Control Panel), 282

Windows Firewall with Advanced Security icon (Control Panel), 278

Windows Future Storage (WinFS), 4

Windows Mail
accounts, configuring, 516-519
filtering, 537-538, 540
incoming, 520-521
customizing columns, 524
processing messages, 522-523
reading, 525-527
managing, 535-537
new features, 29
newsgroups
customizing, 559-562
subscribing, 550-552
searching, 540-542

How can we make this index more useful? Email us at indexes@samspublishing.com
sending, 527-529
applying recipient shortcuts, 530
creating signatures, 530
customizing, 531-534
Windows Marketplace, hardware compatibility, 458
Windows Media Audio (WMA), 204
Windows Media Audio Pro (WMA Pro), 205
Windows Media Player 11, 192-193
album art, 194-195
copying music to/from CDs, 204-208
file types, associating, 200
keyboard shortcuts, 687
Library navigation, 193-194
Media Sharing, 199-200
metadata, 195, 197
new features, 30, 32
playback options, 202-203
playing files, 200-201
searching, 197
security, 605-606
shortcut keys, 201
syncing with media devices, 198-199
views, 194, 196
in Windows XP, 4
Windows Media Player toolbar, 361
Windows Meeting Space, 654
ending, 661
inviting, 657-658
joining, 656
managing, 660
new features, 36
sharing handouts, 658
signing in, 655
starting, 655-656, 658-659
Windows Memory Diagnostic Tool, 21, 61, 455
Windows Mobile Device Center icon (Control Panel), 282
Windows Mobility Center, 35, 249-250
Windows Mobility Center icon (Control Panel), 282
Windows Photo Gallery, 188-189, 191-192
applying tags, 189-190
editing images, 191-192
grouping images, 189-190
new features, 33
searching images, 191
Windows PowerShell, 4
Windows Presentation Foundation (WPF), 4, 23
Windows Registry. See Registry
Windows Resource Exhaustion Detection and Resolution (RADAR), 21
Windows Script Host. See WSH
Windows Search Engine, 17, 85-86
Windows Service Hardening, 22, 606
Windows Sidebar, 18
Windows Sidebar icon (Control Panel), 282
Windows SideShow, 259-260
Windows SideShow icon (Control Panel), 282
Windows System Assessment Tool (WinSAT), 380-381
Windows Update icon (Control Panel), 282, 285
Windows Update website, 451
Windows Vista
applications
adding memory, 391
configuring priority, 393-394
installing fast hard drives, 391
optimizing, 391-392
starting, 392
upgrading device drivers, 392
backups, 424
automating, 425, 427
creating system images, 427
dropped features, 4-5
editions of, 10-12
  upgrading with Windows Anytime Upgrade feature, 12-14
Event Viewer, 430, 432, 434
files, deleting, 415-417
free disk space, checking, 412-414
hard disks, defragmenting, 417-419
hard drives
  closing Indexers, 395
  configuring Last Access Time, 397
  converting partitions to NTFS, 396-397
  disabling compression/encryption, 395
  enabling write caching, 395-396
  implementing maintenance, 394
  optimizing, 394
  turning off 8.3 filename creation, 397
  viewing specifications, 394
history of, 2-3
logon process. See logon process
maintenance
  checking hard disks for errors, 407-409
  running Check Disk, 410-412
new features
  burning DVDs, 33
  Control Panel, 24, 26-27
  Desktop Window Manager (DWM), 23
  Game Explorer, 34-35
  graphics, 23
  interface features, 14-18
  Internet Explorer 7, 27-29
  Media Center, 32
  metadata support, 18-19
  Mobility Center, 35
  Network and Sharing Center, 35
  Network Map, 35-36
  per-application volume control, 33
  performance, 19-20
security, 21-23
Sound Recorder, 34
stability, 20-21
Transactional NTFS (TxF), 23
Welcome Center, 24-25
Windows Backup, 34
Windows Calendar, 29-30
Windows Easy Transfer, 34
Windows Mail, 29
Windows Media Player 11, 30, 32
Windows Meeting Space, 36
Windows Photo Gallery, 33
Windows Presentation Foundation (WPF), 23
XML Paper Specification (XPS), 24
performance improvements, 374
  Restart Manager, 378, 380
  Sleep mode, 375-376
  startup speed, 374-375
  SuperFetch with ReadyBoost, 376-378
performance monitoring, 380
  with Reliability and Performance Monitor, 385-389
  with Task Manager, 382-385
  viewing performance rating, 380-382
redirecting, 747-749
scheduling maintenance, 434-435
stability improvements, 403
  I/O cancellation, 404
  Reliability Monitor, 405
  service recovery, 406
  Startup Repair Tool, 407
startup
  closing splash screens, 390
  configuring prefetchers, 391
  logging on automatically, 391
  optimizing, 389

How can we make this index more useful? Email us at indexes@samspublishing.com
reducing BIOS checks, 389
speeding up, 390
upgrading device drivers, 390
system requirements, 5
graphics, 8-9
hardware, 9-10
memory, 6-7
processors, 5-6
storage, 7-8
System Restore, configuring, 421-423
updates, 428-429
checking for security vulnerabilities, 430-431
virtual memory
customizing page files, 398-399
modifying page files, 399-401
monitoring page files, 399
optimizing, 398
saving page files, 398
splitting page files, 398
Windows Vista Business edition, 11
Windows Vista compatibility
compatibility layers
defined, 144-146
scripting, 145, 147-148
described, 142-143
determining, 143-144
pre-installation checklist, 125
Windows Vista Display Driver Model (WDDM), 9
Windows Vista Enterprise edition, 11
Windows Vista Home Basic edition, 11
Windows Vista Home Premium edition, 11
Windows Vista Starter edition, 12
Windows Vista Ultimate edition, 12
Windows XP
backports, 4
down-level tools, 4
editions of, 10
history of Windows Vista, 3
WINFILE.INI file, 292
WinFS (Windows Future Storage), 4, 65
WinSAT (Windows System Assessment Tool), 380-381
Wired Equivalent Privacy (WEP), 614
wired networks, setup, 609-611
wireheads, 39
wireless hotspots, defined, 613
wireless network setting (Windows Mobility Center), 250
wireless networks
ad hoc wireless networks, connecting to, 617-618
connecting to, 612-615
managing, 624-625
nonbroadcasting wireless networks, connecting to, 615-616
security settings, changing, 625
WMA (Windows Media Audio), 204
WMA Pro (Windows Media Audio Pro), 205
WMP. See Windows Media Player 11
word processors, document comparison in, 129
workgroup names, changing, 611-612
workgroups
remote connections, 663, 665-672
configuring, 664-665
disconnecting, 672
Internet, 673-676
viewing, 635
WPA (Wi-Fi Protected Access), 614
WPF (Windows Presentation Foundation), 23, 4
WPG. See Windows Photo Gallery
write caching, enabling, 395-396
WScript
applying, 316-317
objects, 324
automating, 325-331
WSE. See Windows Search Engine

WSH (Windows Script Host), 313, 315

  Internet Explorer, 345-346, 348-349
  programming objects, 319-324

scripts

  applying CScript, 317
  applying WScript, 316-317
  customizing properties, 318-319
  executing, 315
  running, 315

WScript objects, 324-331

WshNetwork objects, 343-345

WshShell objects, 331-333, 335-341, 343

WshNetwork objects, programming, 343-345

WshShell objects, programming, 331-333, 335-341, 343

X

XCOPY command, 733-734, 745-746

XPS (XML Paper Specification), new features, 24

Z

zombie computers, 567

zones (security)

  adding/removing sites, 575
  configuring, 573-574
  modifying sites, 576