

Index

• Symbols •

- & (ampersand), in LDAP filters, 279–280
 - <> (angle brackets), enclosing required items in syntax, 54
 - <# #> (angle bracket, pound sign), enclosing AutoHelp text, 199
 - * (asterisk)
 - in LDAP filters, 278
 - multiplying numbers, 143
 - in regular expressions, 133–134
 - *= (asterisk, equal), multiplying and assigning numbers, 143
 - @ (at sign)
 - preceding array definition, 46, 149
 - preceding hash table definition, 155
 - preceding Here-Strings, 119–120
 - \ (backslash), in regular expressions, 129
 - ` (backtick), continuation character, 44
 - ^ (caret), regular expression, 131, 134–135
 - { } (curly braces)
 - enclosing expressions to debug, 69–70
 - enclosing functions, 188
 - enclosing variable names, 62
 - \$ (dollar sign)
 - preceding variables, 62, 71
 - in regular expressions, 135
 - \$? variable, 72
 - \$\$ variable, 72
 - \$^ variable, 72
 - \$_ variable, 72
 - . (dot operator)
 - in regular expressions, 129–130
 - between variable and method/property, 75
 - "" (double quotes), enclosing literal strings, 119
 - ! (exclamation point), in LDAP filters, 278–279
 - / (forward slash), dividing numbers, 143–144
 - /= (forward slash, equal), dividing and assigning numbers, 144
 - (hyphen), preceding parameters, 54
 - (minus sign), subtracting numbers, 142
 - (minus sign, double), decrementing numbers, 142
 - = (minus sign, equal), subtracting and assigning numbers, 142
 - % (modulus operator), 144
 - | (pipe character)
 - in LDAP filters, 279–280
 - for piping, 51
 - in regular expressions, 135
 - + (plus sign)
 - adding numbers, 140–142
 - combining strings, 121
 - in regular expressions, 132–133
 - ++ (plus sign, double), incrementing numbers, 141
 - += (plus sign, equal), adding and assigning numbers, 140
 - # (pound sign), preceding comments, 42
 - ? (question mark)
 - debugger command, 241
 - in regular expressions, 132
 - > (redirection operator), 164
 - [] (square brackets)
 - enclosing array indexes, 149, 152–153
 - enclosing casted data types, 68–69, 121
 - enclosing character sets in regular expressions, 130–131
 - enclosing data types in variable definition, 66
 - enclosing optional items in syntax, 54
 - enclosing parameter properties, 203
- ## • A •
- Active Directory (AD)
 - connecting to, 274–275
 - definition of, 273–274
 - filtering queries of, 277–282
 - modifying object attributes, 283–285
 - obtaining distinguishedName for users, 282–283
 - psbase object, accessing, 285–286
 - querying, 275–277
 - updating group membership, 284–285
 - Active Directory For Dummies (Clines, Loughry), 274
 - Active Directory Services Interface (ADSI), 45, 274–275
 - ActiveX Data Objects (ADO), 312–316
 - AD. *See* Active Directory (AD)
 - Add() method, for dates, 180–181
 - add() method, for hash tables, 157
 - Add-Content Cmdlet, 166

- addition of numbers, 140–142
 - Add-Member Cmdlet, 350
 - AddPrinterConnection() method, Win32_Printer class, 341
 - ADO (ActiveX Data Objects), 312–316
 - ADSI (Active Directory Services Interface), 45, 274–275
 - advanced functions
 - attributes for, 199–200
 - AutoHelp text in, 199
 - definition of, 197–198
 - dot-sourcing, 205
 - libraries of, 207
 - methods for, 204
 - parameters for, 200–203
 - in profile script, 205
 - running, 204–206
 - structure of, 198–199
 - as wrapper around a Cmdlet, 207
 - alias drive, 32–33
 - Alias property, for parameters, 200
 - aliases, 30–34
 - AllowEmptyCollection property, for parameters, 202
 - AllowEmptyString property, for parameters, 202
 - AllowNull property, for parameters, 202
 - AllSigned execution policy, 289
 - ampersand (&), in LDAP filters, 279–280
 - AND operator, 77
 - angle bracket, right (>), redirection operator, 164
 - angle brackets (<>), enclosing required items in syntax, 54
 - Anonymous impersonation level, 111
 - append switch, Out-File Cmdlet, 165
 - AppendChild() method, XML object, 168–169
 - application firewall policies, 299–300
 - \$Args variable, 72
 - arguments (parameters) for functions
 - for advanced functions, 200–203
 - \$Args variable containing, 72
 - for functions, 189–191
 - arguments on command line, 41–42
 - array data type, 67
 - arrays
 - accessing elements of, 149–151
 - adding elements to, 151
 - creating, 46, 148–149
 - definition of, 148
 - examples using, 153–154
 - length of, determining, 150
 - looping through, 150–151, 363
 - looping through hash tables using, 158
 - multidimensional, 152–153
 - populating in a loop, 46
 - as operator, 69
 - asterisk (*)
 - in LDAP filters, 278
 - multiplying numbers, 143
 - in regular expressions, 133–134
 - asterisk, equal (*=), multiplying and assigning numbers, 143
 - at sign (@)
 - preceding array definition, 46, 149
 - preceding hash table definition, 155
 - preceding Here-Strings, 119–120
 - authentication levels, WMI, 109–111
 - AuthorizedApplication property, HNetCfg.FwMgr object, 326, 328–329
 - auto-complete for commands (tab expansion), 34–35
 - AutoHelp text, 199
 - automatic variables, 71–74
- B •**
- background color for command shell, changing, 22
 - background jobs
 - creating, 210–211
 - enabling WinRM for, 210
 - retrieving results from, 211–212
 - running remotely, 219–220
 - stopping before finished, 214–215
 - terminating when finished, 214
 - waiting for, 213
 - BackgroundColor property, \$Host.UI.RawUI object, 22
 - backslash (\), in regular expressions, 129
 - backtick (`), continuation character, 44
 - basic data types, 63
 - batch files, 302–306
 - Begin() method, for advanced functions, 204, 206
 - body switch, ConvertTo-HTML Cmdlet, 173
 - books and publications
 - Active Directory For Dummies* (Clines, Loughry), 274
 - SQL For Dummies* (Taylor), 107
 - boolean data type, 63, 67
 - Boolean operators, 77–79
 - braces ({}), enclosing expressions to debug, 69–70
 - enclosing functions, 188
 - enclosing variable names, 62

- brackets ([])
 - enclosing array indexes, 149, 152–153
 - enclosing casted data types, 68–69, 121
 - enclosing character sets in regular expressions, 130–131
 - enclosing data types, 66
 - enclosing optional items in syntax, 54
 - enclosing parameter properties, 203
 - branching statements, 80–83
 - breakpoints
 - disabling, 242
 - enabling, 242–243
 - listing, 241–242
 - removing, 243
 - setting, 238–240
 - buffer size for command shell, 22–23
 - BufferSize property, `$Host.UI.RawUI` object, 23
 - bugs (errors). *See also* exceptions
 - caused by execution policy, 365–366
 - caused by incorrect comparison operators, 370–371
 - caused by incorrect function calls, 366–367
 - caused by incorrect function placement, 367–368
 - caused by incorrect logic, 232–233
 - caused by infinite loops, 88
 - caused by misuse of variable scope, 372–373
 - caused by not casting variables as strings, 369–370
 - caused by pipeline data treated as strings, 368–369
 - caused by pipelines that are too long, 372
 - caused by system errors, 233
 - caused by unexpected input, 232
 - debugging, 233–235, 237–243
 - `$Error` variable containing, 72
 - preventing, 235–237
 - byte data type, 63, 67, 138
- C •
- c (Continue) command, in debugger, 241
 - c (prefixing case-sensitive comparison operators), 80
 - Call authentication level, 111
 - camel case notation, for variable names, 62
 - caret (^), in regular expressions, 131, 134–135
 - Cascading Style Sheets (CSS), 353–355
 - case of strings, changing, 127
 - case-sensitive comparison operators, 80
 - casting (changing data types), 67–70, 121, 145–146, 369–370
 - CERT: drive, 294–295
 - certificate. *See* code-signing certificate
 - certificate store, querying, 294–295
 - char data type, 63, 67
 - character sets, in regular expressions, 130–131
 - ChassisTypes property, `Win32_SystemEnclosure` class, 336–338
 - child elements, XML, 167
 - CIM (Common Information Model), 100
 - CIM Object Manager (CIMOM), 101
 - CIMOM Repository, 101
 - classes, WMI. *See* objects, WMI
 - Clines, Steve (*Active Directory For Dummies*), 274
 - `CmdletBinding` attribute, 199–200
 - Cmdlets. *See also* commands
 - creating. *See* advanced functions
 - definition of, 49–50
 - displaying list of, 51–52
 - getting help on, 52–55
 - naming conventions for, 50–51
 - passing output to another Cmdlet. *See* pipelines
 - running in background, 210–215
 - running remotely, 215, 217–218
 - syntax for, interpreting, 54–55
 - wrapper around, with advanced functions, 207
 - code-signing certificate
 - certificate store, querying, 294–295
 - definition of, 290
 - for profile script, 26
 - requesting from Enterprise CA, 293–294
 - self-signed certificate, creating, 290–292
 - signing scripts, 295–297
 - collections. *See* arrays; hash tables
 - colors in command shell, changing, 22
 - COM objects, 307–310, 362
 - command line arguments, 41–42
 - Command pane, ISE, 19
 - Command property, `Job` object, 211
 - command shell for PSH
 - background color for, changing, 22
 - buffer size for, changing, 22–23
 - foreground color for, changing, 22
 - profile script for, 24–27
 - starting, 16–17
 - window size for, changing, 22–23
 - window title for, changing, 24

- commands. *See also* Cmdlets
 - aliases for. *See* aliases
 - current, determining information about, 73
 - multiple, entering before running, 27
 - stringing together. *See* pipelines
 - tab expansion for, 34–35
- comments, 42
- Common Information Model (CIM), 100
- comparison operators, 79–80, 303–304, 370–371
- computersname parameter, for Cmdlets, 215
- computers, monitoring. *See* hardware, monitoring
- Confirm switch, Out-File Cmdlet, 165
- ConfirmImpact attribute, 200
- Connect authentication level, 111
- console file, name of, 72
- \$ConsoleFileName variable, 72
- constants, 70–71
 - contains operator, 80
- continuation character (.), 44
- Continue (c) command, in debugger, 241
- conventions used in this book, 2–3, 7
- ConvertFrom-StringData Cmdlet, 225, 228
- ConvertTo-CSV Cmdlet, 350
- ConvertTo-HTML Cmdlet, 171–174, 346, 352–353, 354
- ConvertTo-XML Cmdlet, 350
- Coordinated Universal Time (UTC), 182
- Copy-Item Cmdlet, 161–162
- CreateObject() method, converting to PSH, 309–310
- CSCRIPT Cmdlet, 60
- CSS (Cascading Style Sheets), 353–355
- CSV files, importing and exporting, 306, 349–350
- culture, determining, 72, 73, 225, 226.
 - See also* internationalization
- \$Culture variable, 225
- curly braces ({})
 - enclosing expressions to debug, 69–70
 - enclosing functions, 188
 - enclosing variable names, 62
- current directory
 - changing, 160
 - determining, 73, 160
 - not in search path, as security feature, 288
 - specifying for command, 41
- CurrentProfile property, HNetCfg.
 - FwMgr object, 326
- CurrentProfileType property, HNetCfg.
 - FwMgr object, 326
- D •
 - \d (regular expressions shortcut), 131
 - \D (regular expressions shortcut), 131
 - DATA section in scripts, 225, 226–229
 - data types
 - casting (changing), 67–70, 121, 145–146, 369–370
 - definition of, 62–63
 - explicitly defining, 65–67
 - implicitly defining, 64–65
 - integral, 138
 - list of, 63–64
 - nonintegral, 139
 - shortcuts (abbreviated names) for, 67
 - date and time
 - adding to, 180–181
 - creating, 178
 - current, determining, 175–176
 - DST (daylight saving time), 181–182
 - formatting, 176–178
 - subtracting, 179–180
 - time zones, 182–184
 - date data type, 63
 - daylight saving time (DST), 181–182
 - [DBG] : prompt, 239
 - debugging. *See also* errors
 - breakpoints for, 238–240, 241–243
 - commands for, 240–241
 - importance of using, 373
 - network connectivity, troubleshooting, 324–325
 - process of, 233–235
 - specific expressions, 69–70
 - stepping through code, 241
 - decimal data type, 63, 67, 139
 - decrementing numbers, 142
 - Default authentication level, 110
 - Default impersonation level, 111
 - DefaultIPGateway property, Win32_NetworkAdapterConfiguration class, 321, 323
 - DefaultParameterSet attribute, 199, 200
 - Delegate impersonation level, 112
 - Delete() method, Win32_Printer class, 342
 - DeleteSubKey() method, RegistryKey class, 271–272
 - DeleteSubKeyTree() method, RegistryKey class, 271
 - DeleteValue() method, RegistryKey class, 271

- Description property, Win32_
 - NetworkAdapterConfiguration class, 321
 - DetectedErrorState property, Win32_
 - Printer class, 342–343
 - DHCPEnabled property, Win32_
 - NetworkAdapterConfiguration class, 321
 - DHCPLeaseExpires property, Win32_
 - NetworkAdapterConfiguration class, 321
 - DHCPLeaseObtained property, Win32_
 - NetworkAdapterConfiguration class, 321
 - DHCPServer property, Win32_
 - NetworkAdapterConfiguration class, 321
 - directories
 - copying, 161–162
 - creating, 160
 - current directory, changing, 160
 - current directory, determining, 73, 160
 - current directory, not in search path, 288
 - current directory, specifying for command, 41
 - deleting, 161
 - moving, 162
 - renaming, 162
 - DirectoryEntry class, 275–277
 - DirectorySearcher class, 276–277
 - Disable-PSBreakpoint Cmdlet, 242
 - disconnected recordsets, 312–313
 - displayhint switch, Get-Date Cmdlet, 176
 - distinguishedName (DN) property, 274, 282–283
 - division of numbers, 143–144
 - DMTF (Distributed Management Task Force), 100
 - DNSDomain property, Win32_
 - NetworkAdapterConfiguration class, 322
 - DNSDomainSuffixSearchOrder property, Win32_
 - NetworkAdapterConfiguration class, 322
 - DNSHostName property, Win32_
 - NetworkAdapterConfiguration class, 322
 - DNSServerSearchOrder property, Win32_
 - NetworkAdapterConfiguration class, 322
 - do until loop, 87
 - do while loop, 86–87
 - dollar sign (\$)
 - preceding variables, 62, 71
 - in regular expressions, 135
 - dot operator (.)
 - in regular expressions, 129–130
 - between variable and method or property, 75
 - dot-sourcing functions, 197, 205
 - double data type, 63, 67, 139
 - double quotes (“”), enclosing literal strings, 119
 - double-clicking, PSH scripts not run by, 288
 - drive space, monitoring, 247–253
 - DriveInfo class, 250
 - DST (daylight saving time), 181–182
- E •**
- Echo command, converting to Windows PowerShell, 302
 - echo response (ping), ICMP, 328
 - Edit Security permission, 103
 - else statement, 80–83, 224–225, 303–304
 - elseif statement, 81–83, 224–225
 - empty strings, 64, 118
 - Enable Account permission, 103
 - EnableDHCP() method, Win32_
 - NetworkAdapterConfiguration class, 321, 325
 - EnableDNS() method, Win32_
 - NetworkAdapterConfiguration class, 321
 - Enable-PSBreakpoint Cmdlet, 242–243
 - EnableStatic() method, Win32_
 - NetworkAdapterConfiguration class, 321, 325
 - encoding switch, Out-File Cmdlet, 165
 - End() method, for advanced functions, 204, 206
 - Enterprise CA, requesting certificate from, 293–294
 - Enter-PSSession Cmdlet, 216–217
 - environment variables, getting value of, 192
 - eq (equal to) operator, 79, 80
 - \$Error variable, 72
 - errors. *See also* exceptions
 - caused by execution policy, 365–366
 - caused by incorrect comparison operators, 370–371
 - caused by incorrect function calls, 366–367
 - caused by incorrect function placement, 367–368
 - caused by incorrect logic, 232–233
 - caused by infinite loops, 88

errors (*continued*)

- caused by misuse of variable scope, 372–373
- caused by not casting variables as strings, 369–370
- caused by pipeline data treated as strings, 368–369
- caused by pipelines that are too long, 372
- caused by system errors, 233
- caused by unexpected input, 232
- debugging, 233–235, 237–243
- \$Error variable containing, 72
- preventing, 235–237
- escape character (\), in regular expressions, 129
- event logs, 109, 257–259
- Exception class, BC2
- exceptions. *See also* errors
 - definition of, BC2–BC3
 - throwing, BC6–BC7
 - trapping, BC3–BC6
- ExceptionsNotAllowed property, HNetCfg.FwMgr object, 326
- exclamation point (!), in LDAP filters, 278–279
- Exclusive OR (XOR) operator, 78
- Execute Methods permission, 103
- execution policy
 - changing, 26–27, 290
 - default, 27
 - levels for, 289
 - problems caused by, 365–366
 - remote operation affected by, 221
- \$ExecutionContext variable, 72
- exit code from program, determining, 73
- Exit-PSSession Cmdlet, 217
- Export-Alias Cmdlet, 33–34
- Export-CliXML Cmdlet, 170, 350
- Export-Csv Cmdlet, 350

● **F** ●

\$False variable, 72, 73–74

file system

- CSV files, importing and exporting, 306, 349–350
- directories. *See* directories
- files, manipulating, 162
- files, reading, 163–164, 361
- files, writing, 164–166, 361

- HTML files, writing, 171–174, 352–353
- XML files. *See* XML files
- FilesystemObject class, converting to PSH, 311–312
- filters, LDAP, 277–282
- FirewallEnabled property, HNetCfg.FwMgr object, 326, 327–328
- folders (directories)
 - copying, 161–162
 - creating, 160
 - current directory, changing, 160
 - current directory, determining, 73, 160
 - current directory, not in search path, 288
 - current directory, specifying for command, 41
 - deleting, 161
 - moving, 162
 - renaming, 162
- FOR command, converting to Windows PowerShell, 304–306
- for loop, 84–85
- force switch, Out-File Cmdlet, 165
- foreach loop, 42, 85, 304–305
- \$ForEach variable, 72
- Foreach-Object Cmdlet, 363
- foreground color for command shell, changing, 22
- ForegroundColor property, \$Host.UI.RawUI object, 22
- format switch, Get-Date Cmdlet, 176–177
- Format-Table Cmdlet, 46, 94–95, 346, 347–348
- forward slash (/), dividing numbers, 143–144
- forward slash, equal (/=), dividing and assigning numbers, 144
- Full Write permission, 103
- function keyword, 188
- functions
 - adding to ISE menu, 28–29
 - advanced functions. *See* advanced functions
 - defining, 188–189, 367–368
 - defining in global scope, 197
 - definition of, 187–188
 - dot-sourcing, 197
 - overlapping names in, 195–197
 - parameters for, 72, 189–191, 366–367
 - in profile script, 197
 - return values for, 191–193
 - running, 189, 366–367
 - scope of variables in, 193–197, 372–373
- FwMgr object, 297–298

• G •

-ge (greater than or equal to) operator, 80
 Get-ChildItem Cmdlet, 32, 42, 267, 311
 Get-Cmdlet Cmdlet, 51–52
 Get-Content Cmdlet, 163–164, 168, 311–312, 361
 Get-Culture Cmdlet, 226
 Get-Date Cmdlet
 creating dates and times, 178
 current date and time, determining, 175–176
 formatting date and time, 176–178
 GetDirectoryEntry() method,
 DirectoryEntry class, 277
 GetDrives() method, DriveInfo class, 250
 Get-EventLog Cmdlet, 257–259
 Get-ExecutionPolicy Cmdlet, 290
 Get-Help Cmdlet, 52–55, 199, 359–360
 Get-ItemProperty Cmdlet, 267
 Get-Location Cmdlet, 160
 Get-Member Cmdlet, 286, 360
 GetObject() method, converting to PSH, 310
 Get-Process Cmdlet, 94–95, 153–154
 GetProfileByType() method, HNetCfg.
 FwMgr object, 327
 Get-PSBreakpoint Cmdlet, 241–242
 Get-PSDrive Cmdlet, 264–265
 Get-Service Cmdlet, 253–254
 GetType() method, for variables, 66
 Get-UIThread Cmdlet, 226
 GetUTCOffset() method, TimeZone
 class, 184
 GetValue() method, RegistryKey
 class, 266
 Get-WMIObject Cmdlet, 103–105, 251–252, 335, 362
 global scope
 for functions, 197
 for variables in functions, 193–195
 globalization. *See* internationalization
 globally open ports, 298, 329
 GloballyOpenPorts property, HNetCfg.
 FwMgr object, 327, 329
 greater than (-gt) operator, 79
 greater than or equal to (-ge) operator, 80
 group membership, updating, 284–285
 grouping data. *See* arrays; hash tables
 -gt (greater than) operator, 79

• H •

h command, in debugger, 241
 hardware, monitoring
 default printer, setting, 342
 inventory script, 333–336
 printer queue, checking, 343–344
 shared network printer, 341–342
 status of hardware, checking, 340
 status of printer, checking, 342–343
 type of computer, determining, 336–339
 USB drives and memory sticks, detecting, 339
 with WMI, 332–340
 hash tables
 accessing elements of, 155–156
 adding elements to, 157
 creating, 155
 definition of, 155, 157
 looping through, 363
 looping through, using arrays, 158
 removing elements from, 157
 HasMoreData property, Job object, 211
 -head switch, ConvertTo-HTML Cmdlet, 173
 help
 on Cmdlets, 52–55, 359–360
 in ISE, 27
 HelpMessage property, for parameters, 201
 Here-Strings, 119–120
 hives, in Windows registry, 262
 HKCU drive, 265
 HKEY_CLASSES_ROOT hive, 262
 HKEY_CURRENT_CONFIG hive, 262
 HKEY_CURRENT_USER hive, 262, 264, 265
 HKEY_LOCAL_MACHINE hive, 262, 265
 HKEY_USERS hive, 262, 264
 HKLM drive, 265
 HNetCfg.FwMgr object, 297–298, 326–327
 \$Home variable, 72
 host for WSH, setting, 60
 \$Host variable, 22, 72
 \$Host.UI.Raw object, 22–23
 HTML files, writing, 171–174, 352–353
 hyphen (-), preceding parameters in syntax, 54

• I •

i (prefixing case-insensitive comparison operators), 80
 ICMP (Internet Control Message Protocol), 328

- IcmpSettings property, HNetCfg.FwMgr object, 327, 328
 - icons used in this book, 7
 - ID property, Job object, 211
 - Identify impersonation level, 112
 - if/else statement, 80–83, 224–225, 303–304
 - Impersonate impersonation level, 112
 - impersonation levels, WMI, 111–112
 - Import-Alias Cmdlet, 33–34
 - Import-CliXML Cmdlet, 170
 - Import-CSV Cmdlet, 306
 - Import-LocalizedData Cmdlet, 225, 227–229
 - incrementing numbers, 141
 - IndexOf() method, for strings, 125–126
 - infinite loops, 88
 - input
 - getting from user, 38–39
 - unexpected, errors caused by, 232
 - \$Input variable, 72
 - instance methods, WMI, 113
 - int data type, 67
 - integer data type, 63, 138
 - integral data types, 138
 - Integrated Shell Environment. *See* ISE
 - internationalization
 - definition of, 223
 - features for, 225
 - importance of, 224
 - sharing scripts, issues regarding, 229–230
 - using if/else statements, 224–225
 - using separation of data, 225, 226–229
 - Internet Control Message Protocol (ICMP), 328
 - interop assemblies, 309
 - Invoke-WMIMethod Cmdlet, 113
 - IPAddress property, Win32_
 - NetworkAdapterConfiguration class, 322, 323
 - IPEnabled property, Win32_
 - NetworkAdapterConfiguration class, 322, 323
 - IPSubnet property, Win32_
 - NetworkAdapterConfiguration class, 322, 323
 - is operator, 80
 - IsDaylightSavingTime() method, for dates, 182
 - ISE (Integrated Shell Environment)
 - Command pane, 19
 - customizing console, 28
 - getting help in, 27
 - keyboard shortcuts, adding, 28–29
 - menu items, adding, 28–29
 - multiple commands, entering before running, 27
 - Output pane, 19
 - partial scripts, running, 27
 - profile script for, 25
 - Script/Editor pane, 19, 20
 - starting, 18, 27
 - IsMatch() method, RegEx object, 128
 - isnot operator, 80
- **I** •
- jobs, background
 - creating, 210–211
 - enabling WinRM for, 210
 - retrieving results from, 211–212
 - running remotely, 219–220
 - stopping before finished, 214–215
 - terminating when finished, 214
 - waiting for, 213
 - .js file extension, 59
- **K** •
- keyboard shortcuts in ISE, 28–29
- **L** •
- \$LastExitCode variable, 73
 - LDAP (Lightweight Directory Access Protocol)
 - connecting to Active Directory using, 274
 - filters for, 277–282
 - le (less than or equal to) operator, 79
 - length property
 - for arrays, 150
 - for strings, 75
 - less than (-lt) operator, 79
 - less than or equal to (-le) operator, 79
 - Lightweight Directory Access Protocol. *See* LDAP
 - like operator, 80
 - line continuation character (`), 44
 - literal strings, 118–119
 - Location property, Job object, 211
 - logic, incorrect, 232–233
 - logical (Boolean) operators, 77–79

long data type, 63, 67, 138
loops

- for arrays, 150–151, 363
- definition of, 83–84
- do until loop, 87
- do while loop, 86–87
- foreach loop, 42, 85
- for hash tables, 158, 363
- infinite loops, 88
- for loop, 84–85
- while loop, 86

Loughry, Marcia (*Active Directory For Dummies*), 274

-lt (less than) operator, 79

• M •

MACAddress property, Win32_
NetworkAdapterConfiguration
class, 322

makecert.exe file, 290–292

managed objects, 100–101

Mandatory property, for parameters, 200

-match operator, 80, 136

mathematical operations

- on dates, 179–181
- on numbers, 139–144

MD (MKDIR) command, 160

memory sticks, detecting, 339

menus in ISE, 28–29

methods

- for advanced functions, 204
- of variables, using, 74–76
- of WMI objects, invoking, 113

Microsoft Developer Network (MSDN), 106

minus sign (-), subtracting numbers, 142

minus sign, double (--), decrementing
numbers, 142

minus sign, equal (=), subtracting and
assigning numbers, 142

MKDIR (MD) command, 160

Mode property, for items in folders, 43

modulus operator (%), 144

Monad, 12–13

Move-Item Cmdlet, 162

MSDN (Microsoft Developer Network), 106

MTU property, Win32_
NetworkAdapterConfiguration
class, 322

multidimensional arrays, 152–153

multimaster configuration, 274

multiplication of numbers, 143

\$MyInvocation variable, 73

• N •

Name property, Job object, 211

namespaces, 101–102, 106

\$NestedPromptLevel variable, 73

.NET classes

- Cmdlets as, 50
- exceptions thrown by, BC3, BC6
- using in PowerShell code, 250, 374

netsh (network shell), 320

network configuration

- with netsh, 320
- TCP/IP settings, 322–325
- with Win32_
NetworkAdapterConfiguration
class, 320–322

network shell (netsh), 320

New-Alias Cmdlet, 31

New-Item Cmdlet, 30–31, 268

New-ItemProperty Cmdlet, 268–269

New-Object Cmdlet, 309–310, 350, 362

New-PSSession Cmdlet, 219

Next() method, Random object, 145

-noClobber switch, Out-File Cmdlet, 165

nodes, XML, 167

-nomatch operator, 80

None authentication level, 110

nonintegral data types, 139

NOT operator, 78

-notcontains operator, 80

NotificationsDisabled property,

HNetCfg.FwMgr object, 327

-notlike operator, 80

-notmatch operator, 136

null strings, 64, 118

\$NULL variable, 73, 74, 118

numbers

- casting (changing data type of), 145–146
- conversion of, automatic, 145
- data types for, 137–139
- mathematical operations on, 139–144
- overflow problems with, 146
- random, 145
- rounding, 144–145

o •

- o (Step-Out) command, in debugger, 241
- object data type, 63
- objects
 - creating for report data, 350–352
 - saving to XML files, 169–171
- objects, COM, 307–310, 362
- objects, .NET. *See* .NET classes
- objects, WMI
 - accessing, 103–105
 - deleting, 114
 - invoking methods of, 113
 - list of classes, 105–106
 - managed objects, 100–101
 - querying, 106–107
 - setting properties of, 114–115
- OpenRemoteBaseKey() method,
 - RegistryKey class, 266, 269–270
- OpenSubKey() method, RegistryKey class, 266, 269–270
- operating systems, 2
- operators
 - comparison operators, 79–80, 303–304, 370–371
 - logical (Boolean) operators, 77–79
 - mathematical operators, 139–144, 179–181
- OR operator, 78
- OU (Organizational Unit), querying, 43–46
- Out-Default Cmdlet, 93–94
- Out-File Cmdlet, 164–165, 312, 361
- Out-GridView Cmdlet, 346, 348–349
- output, displaying, 38–39
- Output pane, ISE, 19
- overflow problems with numbers, 146

o • p •

- Packet authentication level, 111
- Packet Integrity authentication level, 111
- Packet Privacy authentication level, 111
- param keyword, 190–191
- parameters
 - for advanced functions, 200–203
 - \$Args variable containing, 72
 - for functions, 189–191
- ParameterSetName property, for
 - parameters, 201
- Partial Write permission, 103
- percent sign (%), modulus operator, 144

- period (.), dot operator
 - in regular expressions, 129–130
 - between variable and method or property, 75
- permissions, WMI, 103
- persistent aliases, 33–34
- persistent connection, 218–219
- \$PID variable, 73
- ping (echo response), ICMP, 328
- pipe character (|)
 - in LDAP filters, 279–280
 - for piping, 51
 - in regular expressions, 135
- pipelines
 - current object in, determining, 72
 - definition of, 51, 90–91
 - filtering output of, 94–95
 - stringing commands together using, 91–94
 - too long, causing errors, 372
 - treating data as strings, causing errors, 368–369
 - in Windows Shell, 12–13
- plus sign (+)
 - adding numbers, 140–142
 - combining strings, 120–121
 - in regular expressions, 132–133
- plus sign, double (++), incrementing numbers, 141
- plus sign, equal (+=), adding and assigning numbers, 140
- policy, application firewall, 299–300
- policy, execution
 - changing, 26–27, 290
 - default, 27
 - levels for, 289
 - problems caused by, 365–366
 - remote operation affected by, 221
- popup() method, for WSH shell, 248–249
- Position property, for parameters, 200
- pound sign (#), preceding comments, 42
- Power Tab, 35
- PowerGadgets extension, 356
- PowerShell. *See* Windows PowerShell 2
- PowerShell drives
 - accessing registry with, 264–265
 - deleting registry keys and values with, 271
 - reading registry keys and values with, 267–268
 - renaming registry keys with, 271
 - writing registry keys and values with, 268–269
- PowerShell ISE. *See* ISE (Integrated Shell Environment)

primitive values, 63
 printers
 default, setting, 342
 printer queue, checking, 343–344
 shared network printer, 341–342
 status of, checking, 342–343
 PrinterStatus property, Win32_Printer
 class, 342–343
 private scope, 193–195
 process identifier, determining, 73
 Process() method, for advanced functions,
 204, 206
 profile command, 25
 profile script
 defining advanced functions in, 205
 defining aliases in, 33–34
 defining functions in, 197
 definition of, 24
 determining location of, 25
 execution policy for, 27
 for remote operation, 221
 running locally without signing, 26–27
 signing using certificates, 26
 \$Profile variable, 73
 prompt level, determining, 73
 properties
 of parameters, 203
 of variables, 74–76
 of WMI objects, 114–115
 Provider Write permission, 103
 providers, WMI, 100–101
 psbase object, 285–286
 \$PSCulture variable, 72
 .psdl file extension, 225
 PSDrives. *See* PowerShell drives
 PSH. *See* Windows PowerShell 2
 \$PSHome variable, 73
 \$psISE variable, 28
 .psl file extension, 39, 288
 \$PSUICulture variable, 73
 \$PSVersionTable variable, 73, 74
 Public Key Infrastructure, setting up, 290
 \$Pwd variable, 73, 74

• Q •

question mark (?)
 debugger command, 241
 in regular expressions, 132
 quotes, double (“”), enclosing literal
 strings, 119

• R •

random numbers, 145
 Random object, 145
 Read Security permission, 103
 Read-Host Cmdlet, 38–39
 read-only variables, 70–71
 Receive-Job Cmdlet, 211–212
 recordsets, disconnected, 312–313
 redirection operator (>), 164
 REG_BINARY data type, 263
 REG_DWORD data type, 263
 REG_EXPAND_SZ data type, 263
 REG_MULTI_SZ data type, 263
 REG_QWORD data type, 263
 REG_SZ data type, 263
 RegEx object. *See* regular expression
 registry
 accessing with PowerShell drives, 264–265
 accessing with RegistryKey class, 265–266
 definition of, 261–262
 deleting keys and values in, 270–272
 reading keys and values in, 267–268
 Registry Editor for, 262–263, 270
 renaming keys in, 270–271
 writing keys and values in, 268–270
 Registry Editor, 262–263, 270
 registry key, 262
 RegistryKey class, 265–266, 269–270,
 271–272
 regular expression (RegEx object)
 alternative sequences in, 135
 anchors in, 134–135
 case sensitivity of, 129
 character sets used in, 130–131
 definition of, 127–128
 escape character for, 129
 matching and replacing substrings, 136
 matching literal characters, 128–130
 optional sequences in, 132
 repeating sequences in, 132–134
 special characters in, 129
 relational database, 107
 ReleaseDHCPLease() method, Win32_
 NetworkAdapterConfiguration
 class, 321
 ReleaseDHCPLeaseAll() method, Win32_
 NetworkAdapterConfiguration
 class, 321, 324
 remainder of division, 144
 Remote Enable permission, 103

remote operation

- for background jobs, 219–220
- of Cmdlets, 215, 217–218
 - computername parameter for, 214
- persistent connection for, 218–219
- policies affecting, 221
- profiles used by, 221
- of Windows PowerShell, 215–221
- RemoteAdminSettings property,
 - HNetCfg.FwMgr object, 327
- RemoteSigned execution policy, 27, 289
- remove() method, for hash tables, 157
- Remove-Item Cmdlet, 31–32, 161, 271
- Remove-ItemProperty Cmdlet, 271
- Remove-Job Cmdlet, 214
- Remove-PSBreakpoint Cmdlet, 243
- Remove-PSSession Cmdlet, 219
- Remove-Variable Cmdlet, 70–71
- Remove-WmiObject Cmdlet, 114
- Rename-Item Cmdlet, 31, 162, 271
- Rename-ItemProperty Cmdlet, 271
- RenewDHCPLease() method, Win32_
 - NetworkAdapterConfiguration class, 321
- RenewDHCPLeaseAll() method, Win32_
 - NetworkAdapterConfiguration class, 321, 324
- replace() method, for strings, 76, 125
- replace switch, 136
- reports
 - Cmdlets used for, 346
 - creating custom PowerShell objects for data, 350–352
 - CSV file format, 349–350
 - formatting with Cascading Style Sheets, 353–355
 - HTML file format, 352–353
 - interactive table format, 348–349
 - tabular output, 347–348
 - third-party reporting tools for, 355–356
 - XML file format, 349–350
- #REQUIRES Version 2 command, 39
- reserved (automatic) variables, 71–74
- resources. *See* books and publications; Web site resources
- RestoreDefaults() method, HNetCfg.
 - FwMgr object, 329–330
- Restricted execution policy, 27, 289
- return keyword, 191
- RFC 3687, 280
- RFC 4515, 280
- root element, XML, 167
- round() method, for numbers, 144–145

• S •

- \s (regular expressions shortcut), 131
- \S (regular expressions shortcut), 131
- s (Step) command, in debugger, 241
- sAMAccountName attribute, 274–275
- scope of aliases, 30–31
- scope of variables in functions, 193–197, 372–373
- script scope, 193–195
- Script/Editor pane, ISE, 19, 20
- scripts. *See also* profile script
 - comments in, 42
 - creating, 39–41
 - in current folder, running, 41
 - DATA section in, 225, 226–229
 - execution policy for, 27
 - running in ISE, 20
 - running remotely, 217–218
 - sharing, 229–230
 - signing, 295–297
 - version required for, specifying, 39
- scripts, Windows Shell
 - converting to PowerShell, 302–306
 - running from PowerShell, 55–59
- scripts, WSH
 - converting to PowerShell, 306–316
 - running from PowerShell, 59–60
- search patterns. *See* regular expression (Regex object)
- security
 - certificate from Enterprise CA, requesting, 293–294
 - certificate store, querying, 294–295
 - current directory not in search path, 288
 - execution policies, 288–290
 - PSH scripts not run by double-clicking, 288
 - self-signed certificate, creating, 290–292
 - signatures to run profile scripts, 26
 - Windows Firewall, 297–300
 - for WMI, 103, 109–111
- SELECT statement, 107
- Select-Object Cmdlet, 95, 108, 109, 362–363
- self-signed certificate, 290
- ServiceName property, Win32_
 - NetworkAdapterConfiguration class, 322
- services
 - blocked by firewall, 299
 - configuring, 256–257
 - determining status of, 108–109, 253–254
 - starting, 109, 254–256

- Services property, HNetCfg.FwMgr object, 327
- session
 - persistent, 218–219
 - remote, 216–217
- Set-Alias Cmdlet, 31
- SetAttribute() method, XML object, 168
- Set-AuthenticationSignature Cmdlet, 295–297
- Set-Content Cmdlet, 166
- SetDefaultPrinter() method, Win32_Printer class, 342
- SetDNSDomain() method, Win32_NetworkAdapterConfiguration class, 321
- SetDNSServerSearchOrder() method, Win32_NetworkAdapterConfiguration class, 321, 325
- SetDNSSuffixSearchOrder() method, Win32_NetworkAdapterConfiguration class, 321
- Set-ExecutionPolicy Cmdlet, 26, 290
- SetInfo() method, DirectoryEntry class, 283–284
- Set-Item Cmdlet, 30
- Set-ItemProperty Cmdlet, 269
- Set-Location Cmdlet, 32, 160, 267, 360
- Set-PSBreakpoint Cmdlet, 238, 240
- Set-Service Cmdlet, 257
- SetValue() method, RegistryKey class, 269–270
- Set-Variable Cmdlet, 62, 70–71
- Set-WmiInstance Cmdlet, 114–115
- shared network printer, 341–342
- shell identifier, determining, 73
- \$ShellID variable, 73
- short data type, 63, 138
- signatures for scripts. *See* code-signing certificate
- single data type, 64, 67, 139
- slash, equal (/=), dividing and assigning numbers, 144
- snapin attribute, 200
- Snover, Jeffrey P. (Monad architect), 12
- Sort-Object Cmdlet, 95, 109
- split() method, for strings, 122–123
- SQL For Dummies* (Taylor), 107
- SQL for WMI (WQL), 106–107
- square brackets ([])
 - enclosing array indexes, 149, 152–153
 - enclosing casted data types, 68–69, 121
 - enclosing character sets in regular expressions, 130–131
 - enclosing data types, 66
 - enclosing optional items in syntax, 54
 - enclosing parameter properties, 203
- star operator (*)
 - multiplying numbers, 143
 - in regular expressions, 133–134
- star operator, equal (*=), multiplying and assigning numbers, 143
- Start-Job Cmdlet, 210–211
- Start-Service Cmdlet, 109, 254
- StartService() method, for WMI objects, 255–256
- Start-Sleep Cmdlet, 256
- state of last operation, determining, 72
- State property, Job object, 211
- static methods, WMI, 113
- Step (s) command, in debugger, 241
- Step-Out (o), in debugger, 241
- Step-Over (v), in debugger, 241
- stepping through code in debugger, 241
- Stop-Job Cmdlet, 214–215
- string data type, 64, 67, 121
- strings
 - case of, changing, 127
 - casting variables as, 369–370
 - combining, 120–121
 - comparing, 126
 - definition of, 117–118
 - empty, 118
 - line breaks in, 119–120
 - literal, 65, 118–119
 - null, 118
 - position of specific characters, 125–126
 - searching for patterns in. *See* regular expression (Regex object)
 - special characters in, 119–120
 - splitting, 122–123, 305–306
 - substitutions of, 125
 - substrings of, 123–125
- subtraction of numbers, 142
- SupportShouldProcess attribute, 199
- switch statement, 83
- system errors, 233
- System.DirectoryServices.DirectorySearcher class, 276–277

• T •

- tab expansion, 34–35
- Taylor, Allen (*SQL For Dummies*), 107
- TCP/IP settings, 322–325

Test-Path Cmdlet, 311
 text files. *See* file system
 text strings
 case of, changing, 127
 casting variables as, 369–370
 combining, 120–121
 comparing, 126
 definition of, 117–118
 empty, 118
 line breaks in, 119–120
 literal, 65, 118–119
 null, 118
 position of specific characters, 125–126
 searching for patterns in. *See* regular expression (RegEx object)
 special characters in, 119–120
 splitting, 122–123, 305–306
 substitutions of, 125
 substrings of, 123–125
 Throw keyword, BC6–BC7
 time and date
 adding to, 180–181
 creating, 178
 current, determining, 175–176
 DST (daylight saving time), 181–182
 formatting, 176–178
 subtracting, 179–180
 time zones, 182–184
 time zones, 182–184
 TimeZone class, 183–184
 -title switch, ConvertTo-HTML Cmdlet, 173
 tokens received by shell, determining, 72
 ToLower() method, for strings, 127
 ToUpper() method, for strings, 75, 127
 translation. *See* internationalization
 Trap keyword, BC4–BC6
 troubleshooting. *See* debugging; errors
 \$True variable, 73–74
 Type property, HNetCfg.FwMgr object, 327
 types
 casting (changing), 67–70, 121, 145–146, 369–370
 definition of, 62–63
 explicitly defining, 65–67
 implicitly defining, 64–65
 integral, 138
 list of, 63–64
 nonintegral, 139
 shortcuts (abbreviated names) for, 67

• U •

-uFormat switch, Get-Date Cmdlet, 176–177
 \$UICulture variable, 225
 unexpected input, 232
 Unrestricted execution policy, 289
 USB drives, detecting, 339
 user profile path, determining, 73
 userAccountControl attribute, 280–282
 UTC (Coordinated Universal Time), 182

• V •

v (Step-Over) command, in debugger, 241
 ValidateCount, 202
 ValidateLength, 202
 ValidateNotNull, 203
 ValidateNotNullOrEmpty, 203
 ValidatePattern, 202
 ValidateRange, 203
 ValidateScript, 203
 ValidateSet, 203
 ValueFromPipeline, 201, 205
 ValueFromPipelineByPropertyName, 201
 ValueFromRemainingArguments, 201
 variables
 automatic variables, 71–74
 constants, 70–71
 data types for. *See* data types
 defining, 62, 66
 definition of, 61–62
 environment variables, 192
 naming conventions for, 62
 properties and methods of, using, 74–76
 read-only variables, 70–71
 scope of, in functions, 193–197, 372–373
 .vbs file extension, 59
 version required, specifying in scripts, 39
 vertical bar (|), pipe character
 in LDAP filters, 279–280
 for piping, 51
 in regular expressions, 135
 von Orouw, Marc (developed Power Tab), 35

• W •

\w (regular expressions shortcut), 131
 \W (regular expressions shortcut), 131
 Wait-Job Cmdlet, 213

- WBEM (Web-Based Enterprise Management), 100
- Web site resources
 - Monad Manifesto, 13
 - MSDN (Microsoft Developer Network), 106
 - Power Tab, 35
 - Public Key Infrastructure setup, 290
 - RFC 3687, 280
 - RFC 4515, 280
 - for this book, 2, 7
 - Windows Firewall tools and settings, 327
- Web-Based Enterprise Management (WBEM), 100
- Where-Object Cmdlet, 94–95, 109, 364
- while loop, 86
- width switch, Out-File Cmdlet, 165
- wildcards in LDAP filters, 278
- Win32_Battery class, 338
- Win32_BIOS class, 104, 105, 107
- Win32_ComputerSystem class, 104, 105
- Win32_Directory class, 105
- Win32_DiskDrive class, 335, 339, 340
- Win32_Environment class, 105, 114–115
- Win32_LogicalDisk class, 105
- Win32_NetworkAdapter class, 105
- Win32_NetworkAdapterConfiguration class, 105, 320–322
- Win32_NTLogEvent class, 105, 109, 258–259
- Win32_OperatingSystem class, 105
- Win32_PCMCIAController class, 338–339
- Win32_PerfFormattedData_Spooler_PrintQueue class, 343–344
- Win32_Printer class, 105, 341–343
- Win32_PrintJob class, 105
- Win32_Process class, 105
- Win32_Processor class, 105
- Win32_QuickFixEngineering class, 105
- Win32_Registry class, 106
- Win32_ScheduledJob class, 106
- Win32_Service class, 106, 108–109, 255–256
- Win32_Share class, 106
- Win32_SystemEnclosure class, 336–338
- Win32_TimeZone class, 106
- Win32_Volume class, 251–252
- window size for command shell, changing, 22–23
- window title for command shell, changing, 24
- Windows, versions of, 2
- Windows Firewall
 - application policies, creating, 299–300
 - authorized applications, listing, 328–329
 - authorized services, listing, 299
 - definition of, 325–326
 - disabling, 328
 - enabling, 297, 327
 - enabling inbound ICMP echo requests, 297–298, 328
 - globally open ports, defining, 298
 - globally open ports, listing, 329
 - HNetCfg.FwMgr object for, 326–327
 - managing remotely, 298, 328
 - resetting to default settings, 329–330
 - status of, determining, 297
- Windows Management Instrumentation.
See WMI
- Windows PowerShell 1.0
 - slow adoption of, 13–14
 - uninstalling, 16
- Windows PowerShell 2. *See also* command shell for PSH; ISE (Integrated Shell Environment); *specific topics*
 - alternatives to, 12
 - features in, 14–15
 - history of, 12–14
 - installation path, determining, 73
 - installing, 15–16
 - release covered by this book, 2
 - remote operation of, 215–221
 - specifying version required in scripts, 39
 - starting command shell for, 16–17
 - starting ISE, 18, 27
 - version of, determining, 73
 - versions covered by this book, 7
- Windows PowerShell 2 scripts. *See* scripts
- Windows PowerShell Integrated Shell Environment. *See* ISE
- Windows registry
 - accessing with PowerShell drives, 264–265
 - accessing with RegistryKey class, 265–266
 - definition of, 261–262
 - deleting keys and values in, 270–272
 - reading keys and values in, 267–268
 - Registry Editor for, 262–263, 270
 - renaming keys in, 270–271
 - writing keys and values in, 268–270
- Windows Remote Shell (WinRM), 210, 216
- Windows Scripting Host (WSH)
 - ADO uses, converting to PSH, 312–316
 - COM objects, using in PSH, 307–310
 - converting scripts to PowerShell, 306–316
 - CreateObject() method, converting to PSH, 309–310

Windows Scripting Host (WSH) (*continued*)
function location, converting to PSH, 307
GetObject() method, converting to PSH, 310
I/O, converting to PSH, 311–312
running scripts from PowerShell, 59–60
Windows services, managing, 253–257
Windows Shell scripts
converting to PowerShell, 302–306
running from PowerShell, 55–59
WindowSize property, \$Host.UI.RawUI object, 23
WindowTitle property, \$Host.UI.RawUI object, 24
WinRM (Windows Remote Shell), 210, 216
WMI (Windows Management Instrumentation)
architecture of, 100–101
authentication levels, 109–111
classes, accessing, 103–105, 362
classes, invoking methods of, 113
classes, list of, 105–106
classes, querying, 106–107
classes, setting properties of, 114–115
definition of, 100
event logs, searching, 109

hardware, monitoring, 332–340
impersonation levels, 111–112
namespaces, 101–102
objects, deleting, 114
permissions, 103
services, determining status of, 108–109
services, starting, 109
wmi data type, 67
WMI MMC snap-in, 102–103
WMI providers, 100–101
WQL (WMI Query Language), 106–107
Write-Host Cmdlet, 38, 302
Write-Output Cmdlet, 38–39
WSCRIPT command, 60
WSH. *See* Windows Scripting Host (WSH)



xml data type, 67
XML files
definition of, 166–167
exporting to, 349–350
reading, 168
saving objects to, 169–171
writing, 168–169
XOR (Exclusive OR) operator, 78