

# Index

## SYMBOLS

(!) Not operator, JScript, 48  
\$? (automatic variable), 495  
\$ (dollar sign), 408  
&& (AND) operator, 26  
&&& (logical And) operator, 47–48  
\* (asterisk symbol), 82, 568  
// (double slash), 42, 93–94  
(\_) underscore character, 39  
| pipe sign, 57–58  
|| (Logical Or operator), 47–48  
== (comparison operators), JScript, 45–46

## A

absolute file paths, batch scripts, 99  
access control lists. *See* ACLs (access control lists)  
access modes, opening files, 182  
account lockout policies, 349  
account management utility library, 659–684  
    configuring domain account policies, 677–678  
    examining code for, 659–677  
    functions for renaming and moving accounts, 684  
    managing computers, 682–684  
    managing groups, 678–680  
    managing users, 680–682  
AccountDisabled property  
    computer accounts with LDAP, 453–454  
    user accounts with LDAP, 476–477  
    user accounts with WinNT, 359–368  
accountExpiration function, 682  
ACLs (access control lists)  
    displaying and modifying using CALCS, 815–816  
    IADsACL working with attributes in NDS, 752  
    viewing and setting permissions, 531  
Active Directory. *See also* ADSI (Active Directory Service Interfaces)  
    activities of, 558  
    getting entries with ADSI from, 559–563  
    logical, physical structures of, 291  
    making erroneous changes to, 558–559  
    managing extensions. *See* extensions, Active Directory domain  
    overview of, 557–558  
    reference guide for, 301  
    synchronizing system time, 268

Active Directory Service Interfaces. *See* ADSI (Active Directory Service Interfaces)  
Active Directory Users and Computers, 334, 335  
    -Activity parameter, Write-Progress cmdlet, 501  
Add( ) method, group members, 465  
Add-Content cmdlet, 533–534  
AddDesktop function, 625–627  
AddDesktopURL function, 625–627  
Add-Member cmdlet, 507  
AddPrinter function, 653  
AddPrinterConnection method, 206–208  
    add-PsSnapIn command, 690  
Add/Remove Snap-in dialog box, 293  
AddStartMenu function, 625–627  
AddStartMenuURL function, 625–627  
AddWindowsPrinterConnection method, 206–208  
administrative (default) shares, 414–415  
ADS\_SECURE\_AUTHENTICATION flag, ADSI, 319–321  
ADS\_SECURE\_CREDENTIALS flag, ADSI, 319  
ADSI (Active Directory Service Interfaces), 301–329  
    accessing properties and updating objects, 322–325  
    architecture, 303–306  
    generic object binding, 317–318  
    handling authentication and security, 318–322  
    LDAP provider, 306–309  
    NDS provider, 312–315  
    NWCOMPAT provider, 315–317  
    overview of, 301–302  
    providers, 302–303  
    WinNT provider, 309–312  
    working with IADs methods, 325–329  
ADSI (Active Directory Service Interfaces) core reference, 749–810  
    error codes. *See* error codes, ADSI  
    IADs, 750–752  
    IADsACL, 752  
    IADsADSystemInfo, 753–754  
    IADsBackLink, 754–755  
    IADsCaseIgnoreList, 755  
    IADsClass, 755–757  
    IADsCollection, 757–758  
    IADsComputer, 758–760  
    IADsComputerOperations, 760  
    IADsContainer, 761–762  
    IADsDeleteOps, 762  
    IADsDomain, 762–764  
    IADsExtension, 764

- ADSI (Active Directory Service Interfaces) core reference (*continued*)
  - IADsFaxNumber, 764
  - IADsFileService, 765
  - IADsFileServiceOperations, 765–766
  - IADsFileShare, 766–767
  - IADsGroup, 767–768
  - IADsHold, 768
  - IADsLargeInteger, 768
  - IADsLocality, 769
  - IADsMembers, 769–770
  - IADsNamespaces, 770
  - IADsNetAddress, 770
  - IADsO, 771
  - IADsObjectOptions, 772
  - IADsOctetList, 772
  - IADsOpenDSObject, 773
  - IADsOU, 773–774
  - IADsPath, 774–776
  - IADsPostalAddress, 777
  - IADsPrintJob, 777–778
  - IADsPrintJobOperations, 779–780
  - IADsPrintQueue, 780–782
  - IADsPrintQueueOperations, 782–783
  - IADsProperty, 783–784
  - IADsPropertyEntry, 784–785
  - IADsPropertyList, 785–786
  - IADsPropertyValue, 786–788
  - IADsPropertyValue2, 788–789
  - IADsReplicaPointer, 789
  - IADsResource, 790
  - IADsService, 791–793
  - IADsServiceOperations, 794–795
  - IADsSession, 790–791
  - IADsSyntax, 795
  - IADsTimestamp, 795
  - IADsTypedName, 796
  - IADsUser, 796–802
  - IDirectoryObject, 802
  - IDirectorySearch, 802
  - using, 749–750
- ADSI (Active Directory Service Interfaces), working with AD and PowerShell, 557–579
  - adding to (deleting from) multivalued properties, 578–579
  - administering resources, 402–411
  - administering Windows services. *See* services
  - building library of AD functions, 570–574
  - creating objects, 563, 575–576
  - getting entries from AD with, 559–563
  - introduction to Active Directory, 557–559
  - LDAP filters, 568–569
  - managing local and domain resources, 345–346
  - querying directory services using, 557
  - setting single-valued properties, 576–578
  - using DirectorySearcher. *See* DirectorySearcher object
  - using schema. *See* schema, ADSI
- ADSI Edit utility, 444–447, 575–576
- ADSPATH (Active Directory Service Path), 559–561
- AdsPath string
  - adding and removing group members, 375–376
  - with ADSI LDAP provider, 306–308
  - with ADSI NDS provider, 312–313
  - with ADSI NWCOMPAT provider, 315
  - with ADSI WinNT provider, 309–310
  - binding ADSI objects, 304–306
  - checking group membership, 374–375
  - working with, 322
- aliases, PowerShell
  - changing locations, 526
  - copying, deleting and removing files, 529–530
  - customizing PowerShell library, 690–691
  - ForEach-object cmdlet, 79–80
  - Format-Table, 584
  - Get-Service , 544
  - getting child items, 526–529
  - overview of, 58–60
  - providers, 524–526
  - Stop-Service cmdlet, 545
  - Where-Object cmdlet, 76
- and operator, PowerShell, 80
- applications, passing arguments to, 122
- architecture, ADSI, 303–306
- archives, event log, 244–245
- arguments
  - CScript command-line options, 93–94
  - Dumpel utility, 248–249
  - getting for script, 115–116
  - link shortcut, 216–217
  - passing to applications, 122
  - AT Scheduler, 275
  - system time synchronization, 268
- Arguments property, WshShortcut, 216–217
- arithmetic operators, JScript, 44–45
- arithmetic operators, VBScript, 22–23
- ARP command, 811–812
- arrays
  - JScript, 42–43
  - PowerShell, 73–74
  - PowerShell, formatting, 502
  - VBScript, 21–22
- ASCII text mode, 182–184
- assignment operators, JScript, 47
- ASSOC command, 812
- associations, WMI, 588–589
- associative arrays
  - building PowerShell library with, 704–706
  - defined, 74
  - switching in output, 514–515
  - writing calculated properties as, 503
- Associators of statement, WMI, 588–591
- asterisk (\*) symbol, 82, 568
- AT command, 813

AT Scheduler  
 deleting scheduled jobs, 277–278  
 overview of, 269  
 running CheckFreeSpace as nightly job, 652  
 using in script, 278–279  
 working with, 275–277

AtEndOfLine property, TextStream, 185

AtEndOfStream property, TextStream, 184–185

attachments, scripting for e-mail, 712

ATTRIB command, 814

attribute scoped queries, DirectorySearcher, 573–575

attributes  
 accessing PowerShell, 531  
 Active Directory, 557–558  
 managing schema, 341–342  
 setting and clearing file, 620–621  
 using ATTRIB to display and modify, 814  
 viewing in Properties dialog box, 446–447

Attributes property, File object, 171–173

authentication, ADSI, 318–322

automated jobs, creating, 280–282

automatic (\$?) variable, 495

-autosize switch, format-table, 503, 583–584

AutoUnlockInterval property  
 defined, 349  
 viewing and setting account policies, 352–353  
 WinNT Domain object, 348

## B

/b switch, Dumpel, 250

Back Link attribute, NDS, 754–755

background color, PowerShell library, 688–689

backups, Registry, 232

banner pages, print jobs, 422–423

batch scripts, 52–62  
 command-line options, 94–95  
 documentation and usage, 104–106  
 identifying job name, 97  
 job properties, 102–103  
 overview of, 8, 96–106  
 parsing instructions, 103  
 referencing external objects and type libraries, 100–102  
 script source code, 98–100  
 setting scripting language, 98

binding  
 ADSI objects, 303–306, 317–318  
 to naming contexts, 439–440

bitwise operators, JScript, 49

blank lines, writing to files, 194–195

Bluetooth service, configuring, 548

break statement, JScript, 51

built-in constants, VBScript, 19–20

built-in local groups  
 Builtin domain container for storing, 333  
 creating/modifying group accounts, 373–376  
 defined, 373

built-in system variables, 117–119

built-in user variables, 117–119

buttons  
 adding to message boxes, 134–135  
 evaluating clicks, 136–137

## C

CACLS command, 815–816

calculated properties, 503–504

Call statement, functions, 35

calling functions, JScript, 54–55

calling functions, VBScript, 35

Cancel button, 131–132

CanPauseAndContinue property, Suspend-Service, 547

Case Ignore List attribute, NDS, 755

case-sensitivity  
 comparison operators, 81–82, 535–536  
 constants, 20  
 DirectorySearcher properties, 566–567  
 getting entries from AD with ADSI, 559–560  
 JScript variables, 39–40  
 PowerShell variables, 68  
 strings, 29–30  
 VBScript variables, 13–14

cat command, UNIX, 533

catch statement, JScript, 148–152

CBool () function, VBScript, 17–18

CDate () function, VBScript, 17–18

certificates, obtaining PowerShell, 86–88

ChangePassword () method, 370–372

changePassword function, 681

characters  
 reading from text files, 187–188  
 skipping in text files, 191  
 writing with Write method, 194

checkComputerAccounts function, 683–684

CheckExists function, 620

CheckFreeSpace function, 652–653

checkGroupMembership function, 679–680

CheckMenu function, 623–625

CheckMenu2 function, 623–625

checkRS function, 656–657

checkService function, 654–655

checkUserAccounts function, 682

checkUserGroups function, 682

child items, getting, 526–529

CHKDSK command, 816–817

Choose-NetworkAdapter function, 698

choose-VM function, 691–694

Class property, IADs, 322

\_\_class property, WMI objects, 597

classes  
 Active Directory, 557  
 environment variable, 119–121  
 Hyper-V management, 600  
 WMI, 581

- Clear method, VBScript, 145–146
- ClearArchiveAttribute function, 620–621
- ClearReadOnly function, 620–621
- client, installing ADSI, 303
- Close method, files, 193
- CLSIDs, for Internet Explorer, 100–102
- CMD.EXE, PowerShell vs., 11
- Cmdlets, PowerShell
  - and aliases, 59
  - managing different outputs from, 494–496
  - overview of, 60–61
  - write commands, 493
- CN (common-name identifier), 334
- collections
  - accessing through FileSystemObject, 154
  - defined, 154
  - managing group objects in ADSI using, 331
  - managing group objects with IADsMembers, 769–770
  - managing with IADsCollection, 331–332, 757–758
- color
  - customizing PowerShell library, 688–689
  - VBScript constants, 19
- Column property, TextStream, 185–186
- COM objects
  - ADSI and, 301
  - building PowerShell library using, 706–711
  - sending mail with Microsoft Outlook, 712–714
- Comma Delimited (Text) archived event log, 245
- command-line
  - scripting options, 93–95
  - using aliases at, 59–60
  - using Dumpel at, 249–251
  - using piping to link commands, 57
- command-line utilities, essential, 811–852
  - ARP, 811–812
  - ASSOC, 812
  - AT, 813
  - ATTRIB, 814
  - CACLS, 815–816
  - CHKDSK, 816–817
  - COMPACT, 817
  - CONVERT, 818
  - DATE, 818–819
  - DRIVERQUERY, 819–820
  - EXPAND, 820
  - FC, 821–822
  - FORMAT, 822–823
  - FTP, 823–825
  - FTYPE, 825–826
  - IPCONFIG, 826–827
  - NBSTAT, 827–828
  - NET ACCOUNTS, 828–829
  - NET COMPUTER, 829–830
  - NET CONFIG SERVER, 830
  - NET CONFIG WORKSTATION, 831
  - NET CONTINUE, 831–832
  - NET FILE, 832
  - NET GROUP, 833
  - NET LOCALGROUP, 833–834
  - NET PAUSE, 834–835
  - NET PRINT, 835–836
  - NET SESSION, 836
  - NET SHARE, 836–837
  - NET START, 837–838
  - NET STATISTICS, 838–839
  - NET STOP, 839–840
  - NET TIME, 840
  - NET USE, 841–842
  - NET USER, 842–843
  - NET VIEW, 843–844
  - NETSTAT, 844–845
  - NSLOOKUP, 845–846
  - PATH, 846–847
  - PING, 847–848
  - RECOVER, 848–849
  - ROUTE, 849–850
  - SCHTASKS, 850–851
  - TIME, 851
  - TRACERT, 851–852
- comments
  - following # sign, 84
  - following single quotation marks, 14
  - removing from files before using Select-String, 724
  - skipping lines containing, 191–192
  - using JScript, 42
- common-name identifier (CN), 334
- common-name objects, 334
- Community Extensions, PowerShell, 524
- COMPACT command, 817
- comparison operators
  - JScript, 45–46
  - PowerShell, 81–82
  - for regular expressions, 721
  - VBScript, 24–26
- compression, on NTFS partitions, 817
- computer and user scripts, 289–300
  - alternatives to group policy, 299–300
  - global group policies for, 294–295
  - how policies are used, 290–292
  - local group policies for, 292–294
  - logon and logoff scripts for, 298–299
  - reasons to use, 289–290
  - startup and shutdown scripts for, 296–297
  - using policy consoles, 295–296
  - when policies are applied, 292
- Computer Management console
  - accessing Event Viewer in, 242
  - checking Task Scheduler in Services node of, 265
  - managing Windows services, 377–382
  - viewing open files and user sessions, 403–404

- computers
  - accessing print queues through, 418–420
  - adding or removing using NET COMPUTER, 829–830
  - defined, 449
  - extended functionality with IADsComputerOperations, 760
  - managing services with IADsService, 791–792
  - managing with IADsComputer, 758–760
  - managing with utility library, 682–684
  - querying across computers, 594–596
  - troubleshooting network routing using TRACERT, 851–852
  - working with local properties, 353–356
  - working with service objects via, 382–386
- computers, managing with LDAP provider, 448–451
  - Active Directory properties, 448–451
  - creating/deleting accounts, 451–452
  - enabling/disabling accounts, 454–455
  - moving/renaming accounts, 452–454
- concatenation of strings
  - JScript, 41–42
  - VBScript, 26
- conditional statements, JScript, 49–50
- conditional statements, VBScript, 26–30
  - conditional controls and strings, 29–30
  - Select Case, 28–29
  - using comparison operators with, 24–26
  - using Else and ElseIf, 27–28
  - using If . . . Then, 26–27
- conditions, PowerShell
  - overview of, 80–82
  - using looping to evaluate, 77–80
- configuration containers
  - core LDAP object model, 333
  - defined, 439
  - viewing default, 335–336
- Conf irm parameter, PowerShell cmdlets, 494
- connections
  - mapping printer, 206–208
  - removing printer, 208–209
  - using NET USE to manage remote, 841–842
  - using NETSTAT to display status of, 844–845
  - using PING to determine network, 847–848
  - for viewing event logs, 242–243
- console device, output to, 483–484
- console tree, Event Viewer, 242–244
- Const keyword, VBScript, 20
- constants, VBScript, 19–20
- contacts, Active Directory
  - Contact object, 449
  - creating, 458–459
  - properties, 453–458
- Container property, IADsClass, 337–338
- containers
  - Active Directory, 558
  - checking to see if objects are, 337–338
  - core LDAP object model, 333–336
  - core WinNT object model, 331
  - Domain NC node, 445–447
  - implementing with IADsContainer, 331, 761–762
- Containment property, IADsClass, 337–338
- contains operator, PowerShell, 82, 534–535
- contents, file, 533–534
- contexts, Active Directory, 558
- Continue( ) method, services, 396–402
- continue statement, JScript, 51
- control flow with looping, JScript, 50–53
- control loops, VBScript, 30–34
- conversion functions, VBScript variables, 17–18
- CONVERT command, 818
- ConvertToDateTime script, 700
- ConvertTo-HTML cmdlet, 516–517
- Copy( ) method
  - single files, 179
  - single folders, 167–168
- CopyFile( ) method, 176–177
- CopyFolder( ) method, 165–166
- Copy-item cmdlet, 529–530
- core ADSI reference. *See* ADSI (Active Directory Service Interfaces) core reference
- core object models
  - LDAP, 333–336
  - Windows Script Host architecture, 6–8
  - WinNT, 332–333
- Core snap-in, 12
- CP/M operating system, 523
- Create( ) method
  - computer accounts using LDAP, 451–452
  - contacts using LDAP, 457
  - creating groups using LDAP, 460–462
  - creating objects, 575–576
  - creating organizational units, 467–468
  - creating shared folders, 417–418
  - creating user accounts with WinNT, 368–369
- createComputerAccount function, 682–683
- createDLDistGroup function, 679
- createDLSecGroup function, 679
- createEnabledComputerAccount function, 682–683
- createGDistGroup function, 679
- createGSecGroup function, 679
- createLocalGroup function, 679
- createLocalUser function, 680–681
- CreateObject( ) method, 102, 138–141
- createShare function, 657
- CreateShortcut method, Shell object, 214
- CreateTextFile method, FileSystemObject, 174–175
- createUDistGroup function, 679
- createUSecGroup function, 679
- CScript
  - command-line options for, 93–95
  - defined, 5
  - examining information, 110–111
  - running scripts with, 6, 92–95

cscript command, 6  
 CString () function, VBScript, 17–18  
 CSV files  
     Export - CSV, 517–518  
     Import - CSV, 519–520  
     parsing text and, 536–537  
 -currentActivity parameter, Write-Progress, 501  
 CurrentUserCount property, FileShare object, 416–417

## D

data types

JScript, 40–42  
 PowerShell pipe and, 63–65  
 PowerShell richer, 69–73  
 print queue properties, 424

date

DATE command, 818–819  
 FileTime format, 567–568  
 formatting codes, 487–488  
 VBScript constants, 19–20

DbFileNm value, 233–234

DCs (Domain Controllers), AD, 334, 558–562

debug output, PowerShell, 493–494

-Debug parameter

defined, 494  
 Get-Service, 544

default (administrative) shares, 414–415

default domain naming context, 562

defaultNamingContext property, RootDSE object, 439, 441

defPrinter function, 653

delete() method

deleting computer accounts with LDAP, 452  
 deleting contacts using LDAP, 457  
 deleting groups using LDAP, 462–463  
 deleting objects, 578–579  
 deleting organizational units using LDAP, 469  
 deleting scheduled AT jobs, 277–278  
 deleting shared folders, 417–418  
 deleting shortcuts and options, 225–226  
 deleting single files, 178–179  
 deleting single folders, 167  
 deleting startup options, 227–228  
 deleting user accounts with LDAP, 478–479  
 deleting user accounts with WinNT, 369  
 deleteComputerAccount function, 682–683  
 DeleteFile method, 175–176, 628  
 DeleteFolder method  
     deleting folders, 628  
     deleting menus, 226–227  
     deleting multiple folders, 164–165

deleteGroup function, 679

deleteLocalGroup function, 679

deleteLocalUser function, 680–681

DeleteShortcut function, 628

deleteUser function, 681

deleting

jobs, using scripts, 282–283  
 Registry values, using RegDelete method, 236

dependencies

accessing, 387–390  
 building PowerShell library, 723–727  
 stopping service without stopping, 397  
 troubleshooting service problems, 390–395

DependentServices property, 546

Description property, error code, 151–152

<description> element, XML, 735

desktop

adding shortcuts to, 625–627  
 setting shortcut hotkeys, 217  
 using special folders for shortcuts, 212–213

DHCP (Dynamic Host Configuration Protocol), 239–241

Dim keyword, 14

directories

accessing options with IADsObjectOptions, 772  
 accessing services with IDirectoryObject, 802  
 creating and deleting, 530–531  
 managing shared, with NET SHARE, 836–837  
 querying with IDirectorySearch, 802  
 returning contents using Get-ChildItem, 526–527  
 setting for Dumpel, 248

Directory services, 301–302. *See also* Active Directory; ADSI (Active Directory Service Interfaces) core reference

DirectorySearcher object, 564–574

building library of AD functions, 572–574

creation parameters, 564–565

defining, 564

FileTime format, 567–568

finding related entries in AD, 574–576

LDAP filters, 568–569

methods, 566

properties, 565–567

returning SearchResult object, 566–567

disableUserAccount function, 681

disabling computer accounts, 453–454

disks

checking for errors using CHKDSK, 816–817

formatting with FORMAT, 822–823

recovering information using RECOVER, 848–849

display name, service objects, 382–386

display prompts, input boxes, 131

DisplayConsolePrompt function, 636

DisplayDialog function, 636–637

-DisplayName parameter, Get-Service, 544

distribution groups

creating with utility library, 679

creating/modifying group accounts, 373–376

defined, 373

DLLs (Dynamically Linked Libraries), for system providers, 303

DNS (Domain Name System), using NSLOOKUP, 845–846

Do Until loops, VBScript, 33–34

- Do While loops, JScript, 51
  - Do While loops, VBScript, 32
  - documentation, batch script execution, 104–105
  - dollar sign (\$), 408
  - domain account policies, 356–372
    - overview of, 356
    - preparing to view and set, 349
    - setting with utility library, 677–678
    - viewing and setting, 350–353
    - working with domain objects, 357–358
  - domain containers
    - binding to, 440
    - core LDAP object model and, 333–334
    - creating new, 334
    - defined, 439
  - Domain Controllers (DCs), AD, 334, 558–562
  - domain forests, 291
  - domain local groups, 373–376
  - Domain Name System (DNS), using NSLOOKUP, 845–846
  - Domain NC node, 444–447
  - Domain object, WinNT provider, 331–332, 357–358
  - domain trees, 291
  - domainDNS class, 447–448
  - domain-naming context
    - domainDNS class for domains in, 447–448
    - overview of, 445–447
  - domains
    - Active Directory, 558
    - applying group policies to, 291–292
    - creating security groups with utility library, 679
    - defined, 291
    - deleting groups, 679
    - managing with IADsDomain, 762–764
  - DOS, 57–58
  - dot sourced
    - defined, 84
    - prefixing functions as, 689–690
  - drag and drop, 95
  - Drive collection, 154, 737–738
  - Drive object, 199–201
  - Drive property, FileSystemObject, 156
  - DriveExists method, FileSystemObject, 198–199
  - DRIVERQUERY command, 819–820
  - drives, 197–206
    - checking for, 198–199
    - defining with Get-PSDrive, 524–525
    - examining all system, 202–203
    - formatting with FORMAT, 822–823
    - history of, 523–524
    - mapping network, 204–206
    - network resource library functions for. *See* network resource utility library
    - obtaining information, 197–198
    - PowerShell providers, 524
    - using Drive object, 199–201
  - Drives collection, 197, 202–203
  - Dumpel utility
    - accessing event logs in, 244–245
    - arguments for, 248–249
    - defined, 248
    - using, 249–251
    - working in scripts with, 251–253
  - Dynamic Host Configuration Protocol (DHCP), 239–241
  - Dynamically Linked Libraries (DLLs), for system providers, 303
- ## E
- /e switch, Dumpel, 249
  - echo command, 109–110, 484
  - ElseIf statements, VBScript, 27–28
  - e-mail, accessing using .NET objects, 712–714
  - Email Address attribute, NDS, 764
  - enableUserAccount function, 681
  - enabling computer accounts
    - with LDAP using accountDisabled property, 454–455
    - using enableComputerAccount function, 682–683
  - encoding switch, Get-Content cmdlet, 519
  - Enumerator() method, JScript, 159
  - environment variables, 117–121
    - accessing, 118–119
    - PowerShell variables vs., 65–66
    - understanding, 117–119
    - working with special folders using, 169–170
  - Err object, VBScript, 144–145, 147
  - error() function, 111
  - error codes, ADSI, 802–810
    - LDAP with Win32, 804–807
    - LDAP with Win32 for ADSI 2.0, 807–810
    - overview of, 802–803
    - standard, 803–804
  - Error event, 244–245
  - Error object, JScript, 151–152
  - Error[] automatic variable, 495
  - ErrorAction parameter, PowerShell Cmdlets, 494
  - \$ErrorActionPreference, cmdlets, 494
  - errors, 141–152
    - checking disks using CHKDSK, 816–817
    - examining with GetErrorInfo, 637
    - managing different outputs from cmdlet, 494–496
    - more on outputs, 496–499
    - runtime, checking for, 122–123
    - runtime, in JScript, 147–152
    - runtime, in VBScript, 141–147
    - setting instructions in scripts, 102–103
  - ErrorVariable parameter, PowerShell Cmdlets, 494–495
  - event logs, 242–262
    - archiving, 244–245
    - clearing, 553
    - exporting, 553–554
    - finding entries in, 554–555
    - generating reports, 253–261

- event logs (*continued*)
    - overview of, 242
    - reading, 248–253
    - understanding entries, 243
    - understanding registry structure, 232–233
    - viewing, 242–243
    - writing to, 245–247
  - event types
    - overview of, 244
    - specifying for LogEvent method, 246
  - Event Viewer, 242–243
  - EventLog method, 244–245
  - events
    - using WriteEvent, 638
    - VBScript, 34
    - WshRemote object, 744
  - .EVTx format, 553
  - <example> element, XML, 735
  - exclude parameter, Get-Service, 544
  - Exit Do loops, VBScript, 34
  - Exit For statement, VBScript, 32
  - Exit Function statement, 36
  - EXPAND command, 820
  - ExpandEnvironmentStrings() method, WScript.
    - WshShell, 108, 119
  - explicit outputs, PowerShell, 490–492, 515
  - explicit variables, declaring, 14
  - export- CliXML, 518–519
  - export- cmdlets, 515
  - Export- cmdlets, 517–518
  - Export- CSV, 517–518
  - exporting, event logs, 553–554
  - exposed objects
    - core WSH, 6–7
    - key WSH, 107–109
  - expressions
    - conditions as, 80
    - regular. *See* regular expressions
    - writing calculated properties with, 503
  - extensions
    - ADSI provider, 302, 303
    - displaying and modifying file types used with FTYPE, 825–826
    - online resources for PowerShell, 729
    - using IADsExtension, 764
  - extensions, Active Directory domain, 439–479
    - accessing schema, 444–448
    - managing computer objects with LDAP, 448–455
    - managing contacts with LDAP, 455–459
    - managing groups with LDAP, 459–465
    - managing user accounts with LDAP. *See* user accounts, managing with LDAP
    - working with naming contexts and RootDSE object, 439–444
    - working with organizational units, 466–469
  - external objects, batch scripts, 100–102
  - external scripts, batch scripts, 99–100
- ## F
- f (format) operator, strings followed by, 485
  - Facsimile Telephone Number attribute, NDS, 764
  - Failure Audit event, 244–245
  - FC command, 821–822
  - fields, selecting and sorting in PowerShell, 505–509
  - file associations, displaying and modifying, 812
  - file extensions
    - .PSI, 12
    - saving Windows scripts with appropriate, 5
    - for use with scripting hosts, 8
  - file I/O tasks, handling with utility library, 634–636
  - file services, managing
    - with IADsFileService, 765
    - with IADsFileServiceOperations, 765–766
    - open resources with IADsResource, 790
    - user sessions with IADsSession, 790–791
  - file time, Windows, 567–568
  - File Transfer Protocol (FTP), 824–825
  - FileExists method, FileSystemObject, 173–174, 192
  - FileInfo objects, 72–73
  - files, 170–180. *See also* folders
    - comparing differences between two files with FC, 821–822
    - copying, moving and deleting, 175
    - creating, 173–175
    - creating with NewFile, 627
    - deleting with DeleteFile, 628
    - issues for multiple, 175–178
    - issues for single, 178–180
    - managing open, with NET FILE, 832
    - manipulating with FileSystemObject, 153–156
    - opening, 181–184
    - properties, 170–173
    - reading text. *See* reading text files
    - skipping lines in, 190–192
    - writing to, 192–195
  - files, PowerShell cmdlets, 523–539
    - changing locations, 526
    - copying, deleting and removing files, 529–530
    - creating and deleting, 530–531
    - file properties and attributes, 531
    - getting child items, 526–529
    - parsing text, 536–537
    - providers, 524–526
    - reading content, 533–534
    - selecting strings and working with text data, 534–536
    - using PSDrives, 524–525
    - viewing and setting ACL permissions, 531–533
    - working with Registry vs., 537–539
  - FileService object, 404–407, 414
  - file-system utility library, 607–628
    - adding NewFolder and NewFile functions, 627
    - adding shortcuts to desktop and Start menu, 625–627
    - examining code for, 607–618
    - managing menu options, 623–625
    - setting and clearing file attributes, 620–621

- using `CheckExists`, 620
    - using `DeleteFile`, `DeleteFolder` and `DeleteShortcut`, 628
    - using `GetInfo`, `GetSize`, and `GetType`, 620
    - using `GetSubFolders`, `GetFiles` and `GetFolderContents`, 619
    - using special folders, shortcuts and menus, 621–623
  - `FileSystemObject`. *See* `FSO (FileSystemObject)`
  - filters
    - checking how script was invoked, 727–728
    - examining event log using, 553–554
    - functions vs., 84
    - handling piping, 701
    - LDAP, 568–569
    - overview of, 62
  - Find Printers dialog box, 418
  - `FindAll()` method, `DirectorySearcher` object, 566
  - `FindOne()` method, `DirectorySearcher` object, 566
  - firewalls
    - managing using COM objects, 706–711
    - sending mail with Microsoft Outlook, 712–714
  - first definition wins rule, 511
  - flags
    - creating user accounts using LDAP, 475
    - setting for user accounts using LDAP, 476–478
    - setting group types, 578
    - user account control, 577–578
    - using with `OpenDSObject`, 319–321
    - using with `PutEx()` method, 327
  - `FolderExists` method, `FileSystemObject`, 163–164
  - folders, 153–170. *See also* shared folders
    - checking contents of, 157–159
    - copying, 164
    - creating, 163–164
    - creating with `NewFolder`, 627
    - deleting, 164
    - deleting with `DeleteFolder`, 628
    - `FileSystemObject` manipulating, 153–156
    - moving, 164
    - multiple, 164–166
    - properties, 159–163
    - shortcut, 212–214
    - single, 167–168
    - special, 168–170, 212–213
    - using `Attributes` property, 171–173
  - For Each loops, VBScript, 31, 158–159
  - For in loops, JScript, 50–51
  - For loops
    - JScript, 50, 115–116
    - PowerShell, 78–79
    - VBScript, 30–31, 115–116
  - `ForAppending()` method, 192–195
  - `ForEach-object` cmdlet, 73–74, 79–80, 513
  - foreground color, PowerShell library, 689
  - forests, AD, 558
  - FORMAT command, 822–823
    - `/format` switch, `Dumpel`, 249
    - `format-list` cmdlet, 503
  - formats
    - archived event logs, 244–245
    - changing PowerShell object, 509–512
    - disk or floppy drives, 822–823
    - event logs for viewing, 255–261
    - outputting in specific file, 516–518
    - in PowerShell, 502–505
  - `format-table` cmdlet, 75
    - changing scaling of, 516
    - displaying WMI information with, 583–586
    - finding network adapters, 695–697
    - formatting in PowerShell, 502–505
  - `ForWriting` method, 192–195
  - FreeSpace property, `FileSystemObject`, 199–200
  - `fromFileTime()` method, 568
  - `FSO (FileSystemObject)`
    - accessing objects and collections, 154
    - checking folder contents, 157–159
    - defined, 153
    - `Drive` property, 156
    - managing drives, 198–199
    - methods, 154–155
    - using, 156
  - FTP (File Transfer Protocol), 824–825
  - FTP command, 823–825
  - FTYPE command, 825–826
  - functions
    - building library of AD, 570–572
    - checking how script was invoked, 727–728
    - combining JScript and VBScript, 126–127
    - creating script libraries. *See* script libraries
    - examining script engine properties, 113–114
    - handling piping, 701
    - JScript, 53–54
    - PowerShell, 62, 83–85
    - VBScript, 17–18, 34–36
- ## G
- GC (global catalog) extension, ADSI providers, 302
  - generic object binding, ADSI, 317–318
  - `GenObject`, 308
  - `Get()` method
    - defined, 325
    - using `Multivalued` property with, 345–346
    - using service object properties, 387
    - WMI objects, 599–600
  - GET cmdlets, supporting `-NAME`, 595
  - `Get-Acl` Cmdlet, 531–533
  - `Get-Alias` command, 59
  - `get-AllComputers` function, 683
  - `Get-AuthenticodeSignature`, 87–88
  - `Get-ChildItem` cmdlet, 526–527, 531

- Get-Content cmdlet, 519, 533–534
  - getDomainComputers function, 683
  - GetDrive object, FileSystemObject, 199–200
  - GetDriveInfo function, 651–652
  - GetDriveName method, FileSystemObject, 198
  - GetErrorInfo function, 637
  - Get-EventLog cmdlet, 553–555
  - GetEx( ) method, IADs, 325–326
  - GetFiles function, 619
  - Get-FirewallRule, 708–711
  - GetFolder method, FileSystemObject, 157–159
  - GetFolderContents function, 619
  - GetInfo( ) method, IADs, 325, 327
  - GetInfo function, 620
  - GetInfoEx( ) method, IADs, 325
  - Get-IPconfig cmdlet, 697–704
  - Get-Item cmdlet, 531
  - Get-Location cmdlet, 526
  - Get-Member function, PowerShell, 70, 72
  - Get-NetworkAdapter function, 693–697
  - GetObject method, ADSI, 303–304
  - GetParentFolderName property, 162–163
  - Get-Process cmdlet, 502–503, 549–552
  - Get-PSDrive command, 524–525
  - getResource static method, 737
  - GetResponse function, 637
  - GetScript function, 113–114
  - GetSEInfo function, 113–114
  - Get-Service (Gsv) cmdlet, 543–544
  - getServiceInfo function, 654–655
  - GetSize function, 620
  - GetSpecialFolder method, FileSystemObject, 169–170
  - GetSubFolders function, 619
  - GetType( ) method, 71
  - GetType function, 620
  - Get-WMIObject cmdlet
    - alias for, 582
    - displaying WMI information with, 583–585, 600–601
    - finding network adapters, 693–694, 696
    - querying across computers, 594–596
    - using WMI Query Language, 586–588
  - Global Catalog server, AD, 558–562
  - global groups
    - creating/modifying accounts, 373–376
    - defined, 373
    - managing with NET\_GROUP, 833
    - policies, defined, 291
    - policies, logon and logoff script, 298–299
    - policies, managing, 295–296
  - global variables, JScript, 40
  - group membership
    - checking with LDAP, 464–465
    - checking with WinNT provider, 370–372, 374–375
    - displaying with checkUserGroups, 682
    - implementing with IADsMember, 332
    - managing user accounts with LDAP, 478
      - managing with IADsGroup, 767–768
      - using checkGroupMembership, 679–680
  - group policies
    - alternatives to, 299–300
    - building PowerShell library, 686
    - global, 294–295
    - local, 292–294
    - logon and logoff scripts, 298–299
    - startup and shutdown scripts, 296–297
    - using, 290–292
    - using policy consoles, 295–296
    - view/set domain account policies, 349
    - when to apply, 292
  - Group Policy snap-in, 294, 295–296
  - Group-Object Cmdlet, 506
  - groups
    - creating, 575–576
    - implementing membership, 332
    - managing with IADsCollection, 331
    - managing with utility library, 678–680
    - setting types of, 578
    - using Group-Object, 506
  - groups, managing with LDAP, 459–465
    - AD properties, 459–460
    - adding members, 465
    - checking group membership, 464–465
    - creating, 460–462
    - deleting, 462–464
    - moving, 462–464
    - removing members, 465
    - renaming, 463–464
  - groups, managing with WinNT provider, 372–376
    - adding/removing members, 375–376
    - checking membership, 374–375
    - creating, 373–374
    - understanding Windows, 373
  - Groups( ) method
    - LDAP, 478
    - WinNT, 370–372
  - groupType property, 461
  - GUID property, IADs, 322, 324–325
- ## H
- hash tables
    - building PowerShell library using, 704–706
    - defined, 503
    - managing firewall using COM objects, 706
    - techniques for switching in output, 514–515
  - “Hello World”, 483–484
  - help files
    - manually generating errors in, 147
    - for message boxes, 137–138
  - HELPMMSG sub-command, 543
  - Hold attribute, NDS, 768
  - Host, PowerShell, 687–689

- HostComputer property, FileShare, 416–417
  - hotkeys, link shortcut, 216–217
  - HTML documents
    - convertto-HTML cmdlet, 516
    - creating blank lines in, 194–195
  - Hyper-V
    - case study for WMI, 600–602
    - creating custom objects on demand, 512–514
    - getting Directory entries in PowerShell using, 560
    - overview of, 592
    - syntax, 537
    - for WMI, 592–594
- I**
- IADs interface
    - ADSI reference guide, 750–752
    - defined, 322
    - working with methods of, 325–329
    - working with properties, 322–325
  - IADsAcl interface, 752
  - IADsADSystemInfo interface, 753–754
  - IADsBackLink interface, 754–755
  - IADsCaseIgnoreList interface, 755
  - IADsClass interface
    - ADSI reference guide, 755–757
    - defined, 332
    - working with schema class objects, 336–339
  - IADsCollection interface, 331–332, 757–758
  - IADsComputer interface, 758–760
  - IADsComputerOperations interface, 760
  - IADsContainer interface
    - ADSI reference guide, 761–762
    - creating user accounts with WinNT, 368–369
    - deleting user accounts with WinNT, 369
    - implementing containers with, 331
  - IADsDeleteOps interface, 762
  - IADsDomain interface, 762–764
  - IADsExtension interface, 764
  - IADsFaxNumber interface, 764
  - IADsFileService interface, 765
  - IADsFileServiceOperations interface, 765–766
  - IADsFileShare interface, 766–767
  - IADsGroup interface, 767–768
  - IADsHold interface, 768
  - IADsLargeInteger interface, 768
  - IADsLocality interface, 769
  - IADsMembers interface
    - ADSI reference guide, 769–770
    - checking group membership with LDAP, 464–465
    - implementing membership groups, 332
  - IADsNamespaces interface, 770
  - IADsNetAddress interface, 770
  - IADsO interface, 771
  - IADsObjectOptions interface, 772
  - IADsOctetList interface, 772
  - IADsOpenDSObject interface, 318, 773
  - IADsOU interface, 773–774
  - IADsPath interface, 774–776
  - IADsPostalAddress interface, 777
  - IADsPrintJob interface, 777–778
  - IADsPrintJobOperations interface, 779–780
  - IADsPrintQueue interface, 418–429, 780–782
  - IADsPrintQueueOperations interface, 418–429, 782–783
  - IADsProperty interface
    - accessing, 341–342
    - ADSI reference guide, 783–784
    - defined, 332
    - examining schema for object properties, 342–346
  - IADsPropertyEntry interface, 784–785
  - IADsPropertyList interface, 785–786
  - IADsPropertyValue interface, 786–788
  - IADsPropertyValue2 interface, 788–789
  - IADsReplicaPointer interface, 789
  - IADsResource interface, 790
  - IADsService interface, 404, 791–793
  - IADsServiceOperations interface, 794–795
  - IADsSession interface, 790–791
  - IADsSyntax interface, 332, 795
  - IADsTimeStamp interface, 795
  - IADsTypedName interface, 796
  - IADsUser interface, 796–802
  - ICMP Echo requests, Ping, 704–705
  - icons
    - adding to message boxes, 135–136
    - setting location of shortcut, 216, 218–219
  - IDirectoryObject interface, 802
  - IDirectorySearch interface, 802
  - if statements, JScript, 49
  - If Then statement, VBScript, 145
  - If . . . Else statements, JScript, 50, 148–149
  - If . . . Then statements, VBScript, 26–27
  - IIS (Internet Information Services) extension, ADSI providers, 303
  - Image Management Service, Hyper-V, 600
  - implicit outputs, PowerShell, 490–492, 515
  - implicit variables, declaring, 14
  - import - CliXML, 518
  - Import - cmdlets, 518
  - Import - CSV cmdlet, 519–520, 536–537
  - imports, 518–520
  - include parameter, Get - Service, 544
  - Information event, 244–245
  - initialization statement, 78
  - initializing arrays, in JScript, 43
  - input, PowerShell. *See also* I/O (input and output); I/O utility library
    - overview of, 518–520
    - selecting text, 520–521
    - user, 521–522
  - input boxes, 131–133
  - InputBox function, 36, 131–132
  - interactive mode, running scripts in, 94

interfaces. *See also* ADSI (Active Directory Service Interfaces)  
     core reference  
     ADSI LDAP provider, 308–309  
     ADSI NDS provider, 313–315  
     ADSI WinNT provider, 310–312

Internet Information Services (IIS) extension, ADSI providers, 303

I/O (input and output), 129–141  
     overview of, 129–131  
     using input boxes, 131–133  
     using message boxes. *See* message boxes

I/O utility library, 629–638  
     examining code for, 629–634  
     handling file I/O, 634–636  
     handling other I/O tasks, 636–638

IP address, synchronizing system time, 268

IPCONFIG command, 697–704, 826–827

Is operator, VBScript, 25

IsAccountLocked property  
     using LDAP, 477  
     using WinNT, 359–368

IsMember( ) method, 373–376, 465

IsRootFolder property, 161–162

item-Properties, Registry values as, 538

items, defined, 537

## J

<?job ?> element, XML, 102–103, 736

job element  
     creating batch scripts, 97  
     defined, 96  
     using script element within, 98

<job> element, XML, 736

.js files, 8, 91

JScript  
     case sensitivity of scripting engine, 8  
     creating scripts. *See* scripts  
     files. *See* files  
     folders. *See* folders  
     handling runtime errors in, 147–152  
     input and output, 129–133  
     .js files, 8  
     managing drives, 197–206  
     managing network printers, 206–209  
     objects for Windows scripting, 10  
     opening files, 181–184  
     reading text files. *See* reading text files  
     scripting basics. *See* scripts  
     skipping lines in files, 190–192  
     using pop-up dialog boxes, 137–141  
     writing to file, 192–195

JScript essentials, 39–55  
     conditional statements, 49–50  
     control flow with looping, 50–53  
     operators, 44–49  
     using arrays, 42–43

    using comments, 42  
     using functions, 53–55  
     using strings, 41–42  
     variable data types, 40–41  
     variable naming conventions, 39–40

## K

keys, associative array, 74

keys, Registry  
     creating new, 235  
     deleting, 236  
     reading, 233–234  
     understanding structure, 232–233  
     working with file system vs., 537  
     writing, 234–235

Key-Value Pair exchange, Hyper-V, 512–513

## L

/I switch, Dumpel, 249

language, setting scripting, 98–100

Large Integer type, manipulating, 768

lcase( ) function, 29–30

LDAP (Lightweight Directory Access Protocol)  
     core object model, 333–336  
     error codes, 803  
     error codes with Win32, 804–807  
     error codes with Win32 for ADSI 2.0, 807–810  
     filters, 568–569  
     getting entries from AD with ADSI, 559–562  
     some directory services implementing, 301

LDAP ADSI provider  
     core object model for, 333–336  
     overview of, 306–309  
     scripting extensions. *See* extensions, Active Directory domain

leaf elements, ADSI schema, 332

libraries. *See* script libraries

Lightweight Directory Access Protocol. *See* LDAP (Lightweight Directory Access Protocol)

-like operator, PowerShell  
     checking for basic wildcards, 718  
     wildcard comparison with, 81–82  
     working with text data, 534–535

Line property, TextStream object, 185–186

lines  
     reading from text files, 188–189  
     skipping when reading text files, 191–192  
     writing blank lines to files, 194–195  
     writing to files, 194

link shortcuts  
     creating with CreateShortcut method, 214–215  
     creating with NewShortcut function, 622  
     defined, 214  
     setting arguments, 216–217  
     setting hotkeys, 217

- setting icon locations, 218
    - setting properties, 214
    - setting working directories, 219
  - List switch, Select-String cmdlet, 724
  - ListSpecialFolders function, 621–622
  - .lnk extension, 214
  - local and domain resources, 347–376
    - creating and modifying group accounts, 372–376
    - creating and modifying user accounts, 356–372
    - domain account policies, 347–353
    - local computer properties, 353–356
  - Local Computer Policy, 292–293, 295–296
  - local computers, 353–356, 368–369
  - local group policies
    - defined, 290
    - managing, 292–294
    - managing logon and logoff scripts, 298–299
  - local groups
    - creating/modifying group accounts, 373–376
    - defined, 373
    - managing with NET\_LOCALGROUP, 833–834
    - managing with utility library, 679
  - local jobs, scheduling, 265–268
  - local variables, JScript, 40
  - locality
    - changing drive, 526
    - managing with IADsLocality, 769
  - LockoutObservationInterval property
    - defined, 349
    - viewing and setting account policies, 352–353
    - WinNTDomain object, 348
  - LogEvent method, WshShell, 246
  - logical operators, JScript, 47–48
  - logical structures, Active Directory, 291
  - logoff, managing script, 298–299
  - Logoff Properties dialog box, 298–299
  - logon
    - applying user-related policies during, 292
    - assigning as scheduled task, 300
    - assigning to individual user accounts, 300
    - changing account for Task Scheduler, 266–267
    - executing through shell, 299
    - managing scripts, 298–299
    - to remote computer using WMI, 595–596
    - setting built-in user variables at, 117
  - logs. *See* event logs
  - looping
    - JScript control flow with, 50–53
    - PowerShell, 77–80
    - VBScript control, 30–34
- M**
- /m switch, Dumpel, 249
  - main( ) function, 111
  - MAKECERT utility, 86
  - MandatoryProperties, IADsClass, 337, 338–341
  - MapDrive function, 653
  - MapNetworkDrive method, Network object, 204–205
  - markup tags, XML. *See* batch scripts
  - Match operator, PowerShell, 81–82, 534–535, 721
  - MaxBadPasswordAllowed property, 348–349, 352–353
  - MaxPasswordAge property, 348–351
  - MaxRange property, IADsProperty, 342
  - MaxUserCount property, FileShare object, 416–417
  - Measure-Object cmdlet, 506
  - Measure-object cmdlet, 535
  - members, of PowerShell objects, 70
  - Members( ) method, 464–465
  - menus
    - accessing and listing options, 221–224
    - configuring, 211
    - creating, 220–221
    - deleting, 226–227
    - deleting shortcuts and options, 225–226
    - updating shortcuts and options, 224–225
    - using CheckMenu, CheckMenu2, 623–625
    - using NewMenu, AddMenuOption, 622–623
    - using special folders for shortcuts, 212–213
  - message boxes, 133–141
    - basics of, 133–134
    - buttons, 134–135
    - evaluating button clicks, 136–137
    - help files, 137
    - icons, 135–136
    - overview of, 133
    - pop-up dialog boxes, 137–141
  - methods
    - converting to and from strings in JScript, 42
    - defined, 6
    - defining objects by, 69–73
    - DirectorySearcher object, 566
    - Err object, VBScript, 145
    - FileService object, 404
    - FileSystemObject, 154–155
    - IADs interface, 325–329
    - input, output and error streams, 130–131
    - interface. *See* ADSI (Active Directory Service Interfaces)
      - core reference
      - print queues, 429–430
      - Script.Signer object, 741
      - Service objects, 396–402
      - StdErr stream, 739
      - StdIn stream, 738–739
      - StdOut stream, 739
      - WMI, 599–600
      - WScript, 107–108, 741
      - WScript.WshNetwork, 108
      - WScript.WshShell, 108–109
      - WshArguments collection, 740
      - WshController object, 742

methods (*continued*)

- WshEnvironment object, 742
- WshNamed collection, 740
- WshNetwork object, 743
- WshRemote object, 744
- WshScriptExec object, 745
- WshShell object, 745–746
- WshShortcut object, 746
- WshSpecialFolders object, 747
- WshUnnamed collection, 741
- WshUrlShortcut object, 747
- Microsoft Management Console (MMC), 292, 707–709
- Microsoft Outlook, sending mail with, 712–714
- Microsoft Windows Administrator's Pocket Consultant*, 292
- MinPasswordAge property, 348–351
- MinPasswordLength property, 348–350
- MinRange property, IADsProperty, 342
- MMC (Microsoft Management Console), 292, 707–709
- modifier keys, 217
- more pipe command, 58
- Move method
  - single files, 180
  - single folders, 168
- moveAccount function, 684
- MoveFile method, 177–178, 228
- MoveFolder method, 166
- MoveHere( ) method, LDAP provider
  - computer accounts, 452–454
  - contacts, 457
  - groups, 462–463
  - organizational units, 469
  - user accounts, 478–479
- Move - item cmdlet, 529
- MoveTo( ) method, 579
- mrAccount function, 684
- MsgBox function, 37
- Msgbox function, 133–140
- multiple dimensions, using arrays with, 21
- multiple files
  - CopyFile, 176–177
  - DeleteFile, 175–176
  - MoveFile, 177–178
- multiple folders
  - CopyFolder method, 165–166
  - DeleteFolder method, 164–165
  - issues for, 164–166
  - MoveFolder method, 166
- multiple-line comments, JScript, 42
- Multivalued property, IADsProperty, 342–343, 345–346
- \$MyInvocation, 727

## N

- NAME, supporting wildcards, 595
- Name property, IADs, 322
- <named> element, XML, 736

## names

- obtaining service object, 382–386
- obtaining shared printer, 418
- \_\_NAMESPACE property, WMI objects, 596
- namespaces
  - configuring default WMI, 582–583
  - finding available, 596
  - managing with IADsNamespaces, 770
  - WMI object, 581
- naming contexts
  - binding to, 439–440
  - core LDAP object model and, 333
  - defined, 439
  - using ADSI Edit to access, 444–445
- naming conventions
  - constants, 20
  - renaming computer account with LDAP, 453–454
  - renaming groups using LDAP, 463–464
  - renaming organizational units using LDAP, 469
  - renaming user accounts using LDAP, 478–479
  - variables in JScript, 39–40
  - variables in VBScript, 13–14
- native mode, creating universal groups in, 461
- NBSTAT command, 827–828
- NDS (Novell NetWare Directory Services)
  - defined, 301
  - working with ADSI NDS provider, 312–315
  - working with ADSI NWCOMPAT provider, 315–317
- NDS ADSI provider, 312–315, 331–332
- NET ACCOUNTS command, 828–829
- Net Address attribute, NDS, 770
- .NET assemblies, loading into PowerShell library, 690
- NET commands, history of, 524
- NET COMPUTER command, 829–830
- NET CONFIG SERVER command, 830
- NET CONFIG WORKSTATION command, 831
- NET CONTINUE command, 831–832
- NET FILE command, 832
- NET GROUP command, 833
- NET LOCALGROUP command, 833–834
- .NET objects
  - sending mail using, 713–714
  - using Web client object and XML, 714–718
- NET PAUSE command, 834–835
- NET PRINT command, 835–836
- NET SESSION command, 836
- NET SHARE command
  - accessing shared folders, 414
  - obtaining name of print queue object, 418
  - reference guide, 836–837
- NET START command, 541–542, 837–838
- NET STATISTICS command, 838–839
- NET STOP command, 839–840
- NET TIME command, 267–269, 840
- NET USE command, 841–842
- NET USER command, 842–843

- NET VIEW command, 843–844
- NetBIOS, using NBTSTAT, 827–828
- NET.EXE command, 541–543
- NETSTAT command, 844–845
- network adapters
  - finding, 693–697
  - sorting list of, 696–697
- network address, print queue properties, 424
- network resource utility library, 639–657
  - examining code for, 639–651
  - for services, 654–655
  - using CheckFreeSpace, 652–653
  - using GetDriveInfo, 651–652
  - using MapDrive, 653
  - working with printers, 653–654
- network shares, utility functions for, 656–657
- network utilities, customizing PowerShell library, 693–704
  - finding network adapters, 693–694
  - Get-Ipconfig, 697–704
  - managing firewall using COM objects, 706–711
  - Ping, 703–704
- networks
  - displaying resources using NET VIEW, 844
  - displaying status of connections using NETSTAT, 844–845
  - managing printers, 206–209
  - managing routing tables using ROUTE, 849–850
  - mapping drives, 204–206
  - reconfiguring services through registry, 236–242
  - troubleshooting routing using TRACERT, 851–852
- NewFile function, 627
- NewFolder function, 627
- New-Item cmdlet, 531, 537
- New-ItemProperty Cmdlet, 539
- New-PSDrive cmdlet, 525
- non-exposed objects, core WSH, 7
- Not (!) operator, JScript, 48
- not operator, PowerShell, 80
- NotContains operator, PowerShell, 534
- Notepad, creating Windows scripts in, 5
- NotLike operator, PowerShell, 81, 534–535
- NotMatch operator, PowerShell, 535, 721
- nouns
  - and piping, 701
  - and PowerShell cmdlets, 59
- Novell Netware 3, 315–317
- Novell NetWare Directory Services. *See* NDS (Novell NetWare Directory Services)
- NSLOOKUP command, 845–846
- NTFS, 817, 818
- null objects, in PowerShell, 80–81
- Number property, error code, 151–152
- numbers
  - constant declaration in VBScript, 30
  - JScript data types, 41
  - variables in PowerShell, 68
  - variables in VBScript, 15–16
- NWCOMPAT ADSI provider, 315–317
- O**
  - object classes, AD, 447–448
  - object name, service objects, 382–386
  - object references, 773
  - <object> element, XML, 736
  - objects
    - accessing through FileSystemObject, 154
    - ADSI LDAP provider, 308–309
    - ADSI NDS provider, 313–315
    - ADSI WinNT provider, 310–312
    - binding ADSI, 303–306, 317–318
    - common Active Directory, 447–448
    - common-name, 334
    - core WSH, 6–8
    - creating, 575–576
    - customizing on demand, 512–514
    - deleting from directory store, 761–762
    - examining schema for properties of, 342–346
    - JScript, 10
    - key WSH, 107–109
    - PowerShell, richer types and .NET, 69–73
    - PowerShell vs. WSH, 11
    - service, 382–386
    - types, PowerShell pipe and, 63–65
    - updating in ADSI, 322–325
    - VBScript, 9
  - objects, WMI
    - discovering, 596
    - methods, 599
    - overview of, 581–582
    - properties, 596
    - updating, 596
  - OctetList attribute, NDS, 772–773
  - OID property, IADsProperty, 342
  - OK button
    - input boxes, 131–132
    - message boxes, 134
  - On Error Resume Next statement, VBScript, 141
  - online references
    - backup and restore registry, 232
    - PowerShell blogs, 729
    - PowerShell Community Extensions, 524
    - PowerShell extensions, 729
    - searching information about properties, 576
    - WMI objects, 582
  - open files
    - managing with NET FILE, 832
    - viewing, 403–404
  - open resources, 402–411
    - overview of, 402
    - utility functions for, 656–657
    - viewing in scripts, 404–407
    - viewing open files and user sessions, 403–404
    - working with Resource and Session objects, 407–411
  - OpenAsTextStream method, 183–184
  - OpenDSObject method, 318–322

- opening files, 181–184
  - OpenTextFile method, 182–183
  - operators
    - JScript, 44–49
    - PowerShell comparison, 81–82
    - VBScript, 22–26
  - OptionalProperties, IADsClass, 337–341
  - options, CScript, 93–94
  - or operator, PowerShell, 80
  - organization, managing account with IADs0, 771
  - organizational unit identifier (OU), 334–335
  - organizational units
    - applying group policies to, 291–292
    - creating, 467–468, 575–576
    - creating new domain container with, 334–335
    - defined, 291, 466
    - deleting, 469
    - extensions, 466–469
    - managing with IADsOU, 773–774
    - modifying, 468–469
    - moving, 469
    - properties for, 466–467
    - renaming, 469
  - organizationalUnit object, 449
  - OU (organizational unit identifier), 334–335
  - Out- cmdlets, built-in, 515–516
  - Out-Clipboard cmdlet, 516
  - Out-Default cmdlet, 516
  - Out-File cmdlet, 516–517
  - Out-Host cmdlet, 515
  - Outlook.application COM object, 712–714
  - Out-Null cmdlet, 516
  - Out-Printer cmdlet, 516
  - output, PowerShell. *See also* I/O (input and output); I/O utility library
    - changing formatting of objects, 509–512
    - to console, 483–484
    - controlling formatting, 502–505
    - creating custom objects on demand, 512–514
    - implicit and explicit, 489–492
    - managing from Cmdlets, 494–496, 515–516
    - more on error, 496–499
    - session transcripts, 499–500
    - sorting and selecting fields, 505–509
    - in specific file formats, 516–518
    - switching techniques, 514–515
    - tracking progress, 501–503
    - using strings, 484–489
    - verbose and debug, 492–494
  - Out-String cmdlet, 516
  - OutVariable parameter, PowerShell Cmdlets, 494–495
- P**
- <package> element, XML, 736
  - panels, Event Viewer, 242–243
  - parameters
    - DirectorySearcher creation, 564–565
    - Get-EventLog cmdlet, 552–555
    - NewShortcut function, 622
    - PowerShell commands and, 73
    - Select-String cmdlet, 724
    - standard cmdlet, 494–496
    - Start-Service cmdlet, 546
    - Stop-Service cmdlet, 545
    - Write-Progress cmdlet, 501
  - Parent property, IADs, 322
  - ParentFolder property, 161–162
  - parentheses ( ), 35, 39
  - parsing
    - batch scripts, 103
    - text files, 536–537
  - Partial Replica Set, AD, 558
  - partitions, AD, 558
  - PasswordHistoryLength property, 348–349, 351–352
  - passwords
    - configuring domain account policies, 349, 677–678
    - creating users with WinNT, 368–369
    - managing users with LDAP, 475–476
    - managing users with utility library, 681–682
    - managing with NET ACCOUNTS, 828–829
    - setting user, 370, 578
  - PATH command, 846–847
  - Path property, FileShare object, 416–417
  - \_\_PATH property, WMI objects, 596
  - paths
    - accessing with IADsPath, 774
    - cmdlets, 528–529
    - managing with IADsPathName, 775–776
    - working with special folders using, 169–170
  - \$Paths parameter, 62
  - Pause( ) method, print jobs, 437–438
  - Pause( ) method, PrintQueue object, 429–430
  - Pause( ) method, services, 396–402
  - pauseService function, 655
  - percentagecomplete parameter, Write-Progress, 501
  - permissions, computer and user script, 290
  - physical structures, AD, 291
  - PING command, 703–704
  - PING command, 847–848
  - pipe (|) symbol, 11, 57–58
  - piping
    - chaining commands using, 11, 57–58
    - copying, deleting and removing files using, 530
    - defined, 57
    - functions vs. filters and, 701
    - out- cmdlets using, 515
    - PowerShell, 63–64, 74–77
  - pointers, 159
  - policies. *See* domain account policies; group policies
  - Pop-Location Cmdlet, 690–691
  - PopUp( ) function, 138–141

- pop-up dialog boxes
  - creating, 137–141
  - displaying event log reports in, 255–261
  - displaying with `DisplayDialog`, 636
- positioning, input boxes, 132
- Postal Address attribute, NDS, 777
- PowerShell
  - architecture, 10–12
  - files. *See* files, PowerShell cmdlets
  - inputs, 518–522
  - outputs. *See* output, PowerShell
  - working with WMI. *See* WMI (Windows Management Instrumentation)
- PowerShell, building library, 685–730
  - accessing mail using .NET objects, 712–714
  - adding more to environment, 689–691
  - checking how script was invoked, 727–728
  - customizing environment for, 685–687
  - generic “choose” function, 691–693
  - host, 687–689
  - managing firewall with COM objects, 706–711
  - network utilities, 693–705
  - prompt, 689
  - regular expressions, 718–722
  - script dependencies, 723–727
  - uses for hash tables, 704–706
  - using Web client object and XML with .NET, 714–718
- PowerShell fundamentals, 57–88
  - aliases, 58–60
  - arrays, 73–74
  - cmdlets, 60–61
  - conditions, 80–82
  - functions and filters, 62
  - looping, 77–80
  - objects, types and PowerShell pipe, 63–65
  - pipe, 63–65
  - PowerShell pipe, 74–77
  - providers, 60–61
  - richer types and .NET objects, 69–73
  - scripts, script blocks and functions, 83–85
  - scripts and security, 85–88
  - shells, 57–58
  - snap-ins, 60–61
  - variables, 65–69
- precedence, operator, 23, 45
- Preserve keyword, VBScript, 22
- print device, 424
- print jobs, controlling, 430–438
  - displaying with NET PRINT, 835–836
  - monitoring status of, 433–436
  - pausing and resuming, 437–438
  - properties, 430–433
  - using `IADsPrintJob`, 777–778
  - using `IADsPrintJobOperations`, 779–780
  - using `IADsPrintQueue`, 780–782
  - using `IADsPrintQueueOperations`, 782–783
- print processor, 424
- print queues, 418–429
  - accessing `IADsPrintJob` with, 777–778
  - displaying shared, with NET PRINT, 835–836
  - examining, 418–420
  - general printer information, 423–424
  - managing, 429–430
  - prioritizing print jobs, 424–425
  - properties, 420–421
  - scheduling availability, 425
  - status of, 425–429
  - using banner page, 422–423
  - using `PrintQueue` object, 420–429
  - working with, 421–422
- printers
  - adding connections, 206–207
  - managing shared, with NET SHARE, 836–837
  - network resource library functions for, 653
  - removing connections, 208–209
  - setting default, 206
- Printers collection, 738
- `PrintJobs()` method, `PrintQueue` object, 430
- Priority property, print queue, 424–425
- PrivateData property, Host, 687–688
- procedures, VBScript, 34–37
- processes, 549–552
- profile scripts, 689–691
  - adding more to PowerShell library, 689–691
  - building PowerShell library with, 685–686
  - host, 687–689
  - prompt, 689
- programs, running from within scripts, 121–124
- progress, tracking, 501–503
- prompt function, PowerShell, 689
- properties
  - accessing in PowerShell, 531
  - account lockout policy, 349
  - Active Directory computers, 448–451
  - Active Directory contacts, 453–458
  - Active Directory groups, 459–460
  - Active Directory user objects, 470–474
  - adding to and deleting from multivalued properties, 578–579
  - custom fields, 584–586
  - defined, 6
  - defining objects by, 69–73
  - `DirectorySearcher` object, 565–567
  - drive, 199–200
  - Drives collection, 738
  - Err object, VBScript, 145
  - files, 170–173
  - `FileService` object, 404–405
  - `FileSystemObject`, 156
  - folders, 159–163
  - Get-Process cmdlet, 549–552
  - IADS interface, 322–325
  - `IADsClass`, 337–341

properties (*continued*)

- IADsProperty, 342, 784–785
- IADsPropertyEntry, 784–785
- IADsPropertyList, 785–786
- IADsPropertyValue, 786–788
- IADsPropertyValue2, 788–789
- input, output and error streams, 130–131
- interface. *See* ADSI (Active Directory Service Interfaces)
  - core reference
  - job, for batch scripts, 102–103
  - link shortcuts, 216
  - local computers, 353–356
  - organization units, 466–467
  - password policies, 349
  - PowerShell Host variable, 687–688
  - print jobs, 430–432
  - print queue, 420–425
  - Printers collection, 738
  - Registry, 538
  - Resource objects, 407
  - RootDSE object, 440–444
  - scripts, 92, 113–114
  - services, 386–390
  - Session objects, 407
  - setting single-valued, 574–577
  - shared folders, 415–416
  - StdIn stream, 739
  - task, 274
  - updating WMI objects, 596–599
  - URL shortcut, 220
  - user accounts, 356–368
  - WMI object, 596
  - WScript, 107–108
  - WScript object, 742
  - WScript.WshNetwork, 108
  - WScript.WshShell, 108–109
  - WshArguments collection, 740
  - WshEnvironment object, 743
  - WshNamed collection, 740
  - WshNetwork object, 743
  - WshRemote object, 744
  - WshRemoteError object, 744
  - WshScriptExec object, 745
  - WshShell object, 746
  - WshShortcut object, 746
  - WshSpecialFolders object, 747
  - WshUnnamed collection, 741
  - WshUrlShortcut object, 747
- Properties dialog box, 446–447
- providers
  - CMDlet snap-ins implementing, 12
  - PowerShell, 60–61, 524–525
  - PowerShell variables and, 66
- providers, ADSI
  - accessing properties and updating objects, 322–325
  - ADSI LDAP provider, 306–309
  - ADSI NDS provider, 312–315
  - ADSI NWCOMPAT provider, 315–317
  - ADSI WinNT provider, 309–312
  - binding ADSI objects using, 303–306
  - core LDAP object model, 333–336
  - core WinNT object model, 332–333
  - extensions for, 302–303
  - generic object binding, 317–318
  - handling authentication and security, 318–322
  - overview of, 302–303
  - working with IADs methods, 325–329
  - working with schema class objects, 336–339
- PS console file, and snap-ins, 686
- .pscl extension, 686
- .PSI file extension, 12
- Push-Location cmdlet, 690–691
- Put( ) method
  - creating objects in Active Directory, 562
  - overview of, 325–326
  - using Multivalued property with, 345–346
  - WMI objects, 599–600
- PutEx( ) method
  - overview of, 325–330
  - using Multivalued property with, 345–346, 578–579
- pwdLastSet property, 476

## Q

- Quest, 559
- question mark (?), 82
- Quit method, 112
- quotation mark, double (")
  - using for declaring VBScript constants, 20
  - using for VBScript variable types, 15
  - using strings with, 484–485
- quotation mark, single (')
  - comments following, 14
  - using strings with, 484–485

## R

- /r switch, Dumpel, 249
- Raise method, Err object, 147
- Read method, 187–188
- ReadAll method, 187–188, 190
- ReadCharN function, 635
- ReadFile function, 634–635
- ReadFromKeyboard function, 636
- Read-Host Cmdlet, 521–522
- reading
  - to event logs, 248–253
  - registry keys and values, 233–234
- reading text files, 184–190
  - characters, 187–188
  - entire file, 190

- lines, 188–189
  - preparing to, 184–187
- ReadLine method, 187–189
- ReadLineN function, 635
- RECOVER command, 848–849
- recovery options, services, 381–382
- recurse switch, Get-ChildItem cmdlet, 526–527
- ReDim function, VBScript, 22
- reference guides
  - ADSI. *See* ADSI (Active Directory Service Interfaces)
    - core reference
    - command-line utilities. *See* command-line utilities, essential
    - Windows Scripting API. *See* Windows Scripting API
  - <reference> element, XML, 737
  - References of statement, WMI, 588–591
  - [regex] type accelerator, 722
  - Registry. *See* Windows Registry
  - Registry Editor, 100–102
  - RegRead method, WshShell, 233–234
  - regular expressions
    - building PowerShell library, 718–722
    - script dependencies and, 723–728
  - RegWrite method, WshShell, 234–235
  - relative file paths, batch scripts, 99
  - remote scripts, 124–126
  - remote systems
    - logon using WMI to, 595–596
    - managing remote connections with NET USE, 841–842
    - permissions to schedule tasks on, 275
    - scheduling jobs on, 265–268
  - Remove( ) method, 465
  - Remove-item cmdlet, 529–531
  - RemoveNetworkDrive method, Network object, 205–206
  - RemovePrinterConnection method, Network object, 208–209
  - Remove-PSDrive cmdlet, 525
  - rename command, CMD.EXE, 530
  - renameAccount function, 684
  - Rename-Item cmdlet, 530
  - repeated statements, for loops, 78
  - Replica Pointer attribute, NDS, 789
  - reports, event log, 253–261
    - creating logs, 253–255
    - formatting logs for viewing, 255–261
  - <resource> element, 105–106, 737
  - resources
    - managing open. *See* open resources
    - using NET VIEW to display network, 844
  - Restart-Service cmdlet, 546–554
  - Resume( ) method, print jobs, 437–438
  - resumeService function, 655
  - RO (read-only) status
    - defined, 356
    - NT user properties, 357–358
    - using service object properties, 386–387

- RootDSE object
  - accessing System container, 446
  - binding to naming contexts, 440
  - binding to root of directory tree with LDAP, 318
  - getting entries from AD with ADSI, 561–562
  - overview of, 333
  - properties, 440–444
- ROUTE command, 849–850
- router, ADSI, 303
- rules, firewall, 709–711
- Run( ) method, WScript.Shell, 121–124
- Run( ) method, WshShell, 251–253, 278–279
- runtime
  - batch script documentation of, 104–105
  - checking for errors, 122–123
  - handling errors in JScript, 147–152
  - handling errors in VBScript, 141–147
- <runtime> element, 735
- RW (read-write) status
  - defined, 356
  - NT user properties, 357–358
  - service properties, 386–387
  - WinNT Domain object properties, 348

## S

- SAMAccountName property, 575–576
- scheduled tasks, 265–287
  - creating scheduling manager script using, 283–287
  - deleting jobs using scripts, 281–282
  - managing using SCHEDULETASKS, 850–851
  - overview of, 265–267
  - startup and logon scripts as, 300
  - using AT command, 278–279, 813
  - using AT Scheduler, 275–278
  - using automated job creation, 280–281
  - using Task Scheduler wizard, 269–274
- Scheduled Tasks folder, 273–274
- schema, AD, 444–448
- schema, ADSI, 331–346
  - accessing with IADsProperty, 341–342
  - core LDAP object model, 333–336
  - core WinNT object model, 332–333
  - examining object properties, 342–346
  - managing attributes with IADsProperty, 783–784
  - managing classes, 336–341, 755–757
  - managing syntax with IADsSyntax, 795
  - overview of, 331–332
- schema containers, 333, 439
- Schema property, IADs, 322
- SCHEDULETASKS command, 850–851
- scope
  - JScript variable, 40
  - PowerShell variable, 66
- script blocks, PowerShell, 83
- script element, batch scripts, 96

- script libraries
  - account management utility. *See* account management utility library
  - file-system utility. *See* file-system utility library
  - I/O utility. *See* I/O utility library
  - network resource utility. *See* network resource utility library
  - PowerShell. *See* PowerShell, building library
- <script> element, XML, 98, 737
- ScriptEngine( ) function, 113–114
- ScriptEngineBuildVersion( ) function, 113–114
- ScriptEngineMajorVersion( ) function, 113–114
- ScriptEngineMinorVersion( ) function, 113–114
- ScriptFullName( ) property, 113–115
- scripting engines
  - defined, 5
  - FileSystemObject as extension of, 154
  - properties, 113–114
  - in WSH architecture, 8–9
- scripting hosts
  - examining information related to, 110–111
  - Windows Script Host architecture, 8
  - WScript or CScript components, 5
- ScriptName( ) property, 113–115
- scripts
  - batch. *See* batch scripts
  - checking how script was invoked, 727–728
  - combining JScript and VBScript, 126–128
  - displaying text strings, 109–110
  - environment variables, 117–121
  - examining information, 110–117
  - key WSH objects, 107–109
  - overview of, 91–92
  - and PowerShell, 83–88
  - running, 91–95
  - running programs, 121–124
  - running programs from within, 121–124
  - running remotely, 124–126
  - working with Dumpel, 251–253
- scripts, scheduling jobs with
  - automatically, 280–282
  - deleting jobs, 282–283
  - managing, 283–287
  - using AT, 278–279
- Script.Signer object, 741
- searches, 385–386
- SearchResult object, DirectorySearcher, 566–567
- Search-Root directory, 573–574
- Secondsremaining parameter, Write-Progress, 501
- security
  - ADSI, 318–322
  - groups, 373–376
  - running scripts remotely, 124–126
  - scripts in PowerShell and, 85–88
- Security snap-in, 12
- Select Case statement, VBScript, 28–29, 51–52
- Select Program to Schedule dialog box, 270
- Select-Object cmdlet, 505
- Select-String cmdlet, 520, 722–727
- servers
  - displaying statistics using NET STATISTICS, 838–839
  - showing status of DNS using NSLOOKUP, 845–846
- services, 377–402
  - checking status and dependencies, 390–395
  - configuration settings for common, 378–379
  - configuring, 548
  - displaying configuration information with NET CONFIG SERVER, 830
  - extending with FileService object, 404–407
  - with IADsFileService, 791
  - with IADsFileServiceOperations, 794–795
  - with IADsService, 791–793
  - listing network, 837–838
  - managing in Services node, 377–378
  - pausing, 396–402, 834–835
  - properties, 386–390
  - recovery settings, 381–382
  - restarting, 546–547
  - resuming, 547–548, 831–832
  - starting, 396–402, 546, 837–838
  - stopping, 396–402, 545–546, 839–840
  - suspending, 547–548
  - using, 382–386
  - utility functions, 654–655
  - viewing and setting information, 395–396
  - working with, 541–545
- Services console, Task Scheduler, 266
- Services node, Computer Management console, 377–378
- ServicesDependedOn property, 546
- session transcripts, 499–500
- sessions. *See* user sessions
- Sessions node, Computer Management console, 403
- Set keyword, pop-up dialog boxes, 138
- setAccountLockoutInfo function, 678
- Set-Acl cmdlet, 531–533
- SetArchiveAttribute function, 620–621
- Set-Content cmdlet, 533–534
- SetDefaultPrinter method, Network object, 206
- SetHiddenSystem function, 620–621
- SetInfo( ) method, IADs, 325, 327, 329
- Set-ItemProperty cmdlet, 538–539
- Set-Location cmdlet, 526
- setMinPasswordLength function, 677–678
- SetNormalAttribute function, 620–621
- setpassword( ) method, 370, 578
- setPasswordAge function, 677–678
- setPasswordHistory function, 678
- SetReadOnly function, 620–621
- Set-Service cmdlet, 548
- shared folders
  - creating and deleting, 417–418
  - managing with IADsFileShare, 766–767
  - properties, 415–416
  - working with, 413–415

- shell scripts
    - executing logon scripts through, 299
    - in PowerShell, 57–58
    - Windows scripts vs., 5
  - shortcuts
    - adding to desktop and Start menu, 625–627
    - building, 214–220
    - creating with `NewShortcut` function, 622
    - customizing for PowerShell library, 686–687
    - deleting, 225–226, 628
    - updating, 224–225
    - using special folders for, 212–213
  - shutdown, 296–297
  - SimpleMatch switch, `Select-String`, 722
  - single files
    - using `Copy`, 179
    - using `Delete`, 178–179
    - using `Move`, 180
  - single folders, 167–168
  - single-line comments, `JScript`, 42
  - sites
    - applying group policies to, 291–292
    - defined, 291
  - sizing arrays, `JScript`, 43
  - sizing arrays, `VBScript`, 21
  - `SkipLine` method, 191–192
  - skipping lines, files, 190–192
  - `Skip(x)` method, 191
  - SMTP server, sending mail with, 713–714
  - snap-ins
    - customizing PowerShell library, 686, 690
    - PowerShell, 11–12, 60–61
  - `Sort-Object` cmdlet, 506–607
  - special folders
    - adding shortcut to, 622–623
    - displaying list of available, 621–623
    - using for shortcuts, 212–213
    - working with, 168–170
  - `SpecialFolders` method, `WScript.WshShell`, 213
  - `split()` method, text strings, 692
  - `src` attribute, batch scripts, 98–100
  - stacks, PowerShell, 690–691
  - `standalone` attribute, `<?XML ?>` element, 103
  - standard input, 57–58
  - standard output, 483–484
  - `Start()` method, services, 396–402
  - Start menu, adding shortcuts to, 625–627
  - `Start-Service` cmdlet, 546
  - `startService` function, 655
  - `StartTime` property, print queue, 425
  - `Start-Transcript` cmdlet, 500
  - startup applications
    - adding options to, 227–228
    - applying computer-related policies during, 292
    - assigning as scheduled task, 300
    - defined, 211
    - managing computer scripts, 296–297
    - moving options, 228
    - setting shortcut hotkeys, 217
    - using special folders for shortcuts, 212–213
  - Startup Properties dialog box, 298–299
  - status
    - monitoring print job, 433–436
    - monitoring print queue, 425–429
    - troubleshooting service problems, 390–395
    - Statuswhich parameter, `Write-Progress` cmdlet, 501
  - `StdErr` stream, 130–131, 739
  - `StdIn` stream, 130–131, 738–739
  - `StdIn.ReadLine()` method, 368
  - `StdOut` stream, 130–131, 739
  - `Step` keyword, `For` loops in `VBScript`, 31–32
  - `Stop()` method, services, 396–402
  - `Stop-Process` cmdlet, 551
  - `Stop-Service` cmdlet, 545–546
  - `stopService` function, 655
  - `Stop-Transcript` cmdlet, 500
  - streams, 130–131
  - strings
    - displaying, 109–110
    - `JScript`, 41–42
    - output to console using, 484
    - selecting, 534–536
    - using single and double quotes in, 484–485
  - strings, `VBScript`
    - conditional controls and, 29–30
    - declaring constants, 20
    - performing operations on, 26
    - variable types, 15–16
  - subnets, 291
  - subroutines, `VBScript`, 34, 37
  - `SUBST` command, 523
  - Success Audit event, 244–245
  - `Suspend-Service` cmdlet, 546–554
  - `switch` case statement, `JScript`, 51–52
  - `switch` statement, 514–515
  - syntax, managing with `IADsSyntax`, 795
  - `Syntax` property, `IADsProperty`, 342–343
  - System container, 445–446
  - system information, `IADsSystemInfo` interface, 753
  - system providers. *See* providers, ADSI
- ## T
- /t switch, `Dumpel`, 249
  - Tab Delimited (Text) archived event log, 244–245
  - tab expansion, PowerShell, 76
  - target object property, 496
  - `TargetPath` property, URL shortcuts, 220
  - Task Scheduler
    - defined, 265
    - scheduling basics, 265–267
    - synchronizing system time, 267–269

- Task Scheduler Wizard, 269–274
    - changing task properties, 274
    - overview of, 269
    - running, 269–273
    - synchronizing system time, 267–269
    - viewing tasks, 273–274
  - tasks. *See* scheduled tasks
  - TCP/IP
    - using IPCONFIG, 826–827
    - using NBTSTAT, 827–828
  - Tee-Object cmdlet, 495–496
  - Test-Path cmdlet, 528
  - Text (Comma Delimited) archived event log, 245
  - Text (Tab Delimited) archived event log, 244–245
  - text files
    - displaying strings, 109–110
    - input. *See* input, PowerShell
    - output. *See* output, PowerShell
    - parsing, 536–537
    - PowerShell variables, 67–68
    - reading. *See* reading text files
    - selecting strings and working with data, 534–536
    - using regular expressions, 718–722
    - writing to, 192–195
  - TextStream object, 184, 193
  - third-party products, managing AD with, 559
  - Throw keyword, 497–498
  - throw statement, JScript, 148–149
  - time
    - creating PowerShell library, 700
    - displaying with NET TIME, 840
    - using TIME command, 851
    - VBScript constants, 19–20
  - TIME command, 851
  - Timestamp attribute, NDS, 795
  - titles
    - input box, 131
    - message box, 134
  - toString( ) method, 485–486
  - TRACERT command, 851–852
  - Trap keyword, 497–499
  - trees, Active Directory, 558
  - try...catch statements
    - JScript error-handling, 147–149
    - managing groups with utility library, 679
  - tryTest function, JScript, 148–149
  - TYPE command, CMD.EXE, 533
  - type libraries, batch scripts, 100–102
  - Typed Name attribute, NDS, 796
- ## U
- ucase( ) function, 29–30
  - UI property, Host, 687–688
  - UNC (Universal Naming Convention), 524
  - underscore (\_) character, 39
  - universal groups
    - creating in native mode, 461–462
    - creating/modifying group accounts, 373–376
    - defined, 373
  - Universal Naming Convention (UNC), 524
  - Universal Resource Locators (URLs), creating batch scripts, 99
  - UNIX, implementation of pipe, 57–58
  - unlockUserAccount function, 681
  - <unnamed> element, XML, 736
  - UntilTime property, print queue, 425
  - Update-FormatData cmdlet, 511–512
  - updates
    - progress, 501
    - shortcut and menu option, 224–225
    - WMI object, 596–599
    - working with IADs properties, 322–325
  - Update-TypeInfo cmdlet, 501
  - .url extension, 214
  - URL shortcuts
    - creating in current working directory, 215
    - creating with NewShortcut function, 622
    - defined, 214
    - setting properties, 220
  - URLs (Universal Resource Locators), creating batch scripts, 99
  - <usage> element, XML, 105, 736
  - user accounts, 356–372
    - checking group membership, 370–372
    - control flags, 577–578
    - creating with WinNT, 368–369
    - deleting with WinNT, 369
    - managing with library utilities, 680–682
    - managing with NET ACCOUNTS, 828–829
    - managing with NET USER, 843
    - managing with WinNT, 368
    - overview of, 356
    - setting and changing passwords, 370
    - user properties for WinNT, 356–359
    - working with properties, 359–368
  - user accounts, managing with LDAP, 470–479
    - creating, 474–476
    - moving, renaming and deleting, 478–479
    - properties for, 470–474
    - setting flags, 476–478
    - viewing group membership, 478
  - user input, 521–522
  - user scripts. *See* computer and user scripts
  - user sessions, 402–411
    - listing or disconnecting using NET SESSION, 836
    - overview of, 402
    - utility functions for, 656–657
    - viewing in scripts, 404–407
    - viewing open files and user sessions, 403–404
    - working with Resource and Session objects, 407–411
  - UserFlags property
    - enabling/unlocking user accounts, 364–368

- summary of, 361–364
- working with user accounts, 359

UserMustChangePassword function, 681

users

- core WinNT object model, 332–333
- creating, 575–576
- creating user accounts. *See* user accounts
- defined, 449
- managing accounts with IADsUser, 796–802
- managing sessions with IADsSession, 790–791
- scheduling tasks to be run by designated, 273

utility libraries. *See* script libraries

## V

values

- associative array, 74
- converting input, 132–133
- setting for input boxes, 132

values, Registry

- configuring DHCP, 240–243
- configuring WINS, 237–239
- creating new keys and assigning, 235
- deleting, 236
- reading keys, 233–234
- understanding, 232–233
- working with, 538–539
- writing keys and, 234–235

var keyword, 40, 138

variables

- JScript, 39–41
- VBScript, 15–18

variables, PowerShell. *See also* environment variables

- conditions as, 80
- Host , 687–689
- overview of, 65–69

variant data type, VBScript, 15

.vbs files, 8, 91

VBScript

- creating scripts. *See* scripts
- folders. *See* files; folders
- handling runtime errors, 141–147
- input and output. *See* I/O (input and output)
- managing drives, 197–206
- managing network printers, 206–209
- objects for Windows scripting, 9
- opening files, 181–184
- reading text files. *See* reading text files
- scripting basics. *See* scripts
- skipping lines in files, 190–192
- .vbs files, 8
- writing to file, 192–195

VBScript, essentials

- conditional statements, 26–30
- control loops, 30–34
- converting variable types, 17–18

- declaring variables, 14–15
- operators, 22–26
- using procedures, 34–37
- variable naming, 13–14
- variable types, 15–16
- working with arrays, 21–22
- working with constants, 19–20

verbose output, PowerShell, 492–494

-Verbose parameter, PowerShell, 494

verbs, PowerShell cmdlets, 59

version attribute, <?XML ?> element, 103

version information, testing scripts for, 110

view pane, Event Viewer, 242–244

viewDetailedRS function, 656–657

viewing

- folder contents, 157–159
- resource and session usage, 408–411
- resources and session objects, 404–407

viewShareInfo function, 657

Virtual Machines, 592, 600–601

Virtual Switch Management Service, Hyper-V, 600

## W

Warning events, 244–245

warning messages, Outlook, 713–714

WebClient class, 714–718

WhatIf switch, PowerShell, 77

Where-Object cmdlet

- filtering child items with, 528
- pipng and, 75–76
- selecting strings and working with text data, 534–536
- sorting and selecting fields, 505–507
- writing queries in WQL, 587

while loops, JScript, 51

while loops, PowerShell, 78–79

While...WEnd loop, VBScript, 34

wildcards

- copying multiple files, 177
- copying multiple folders, 166
- deleting multiple files, 176, 628
- deleting multiple folders, 165, 628
- Get-VM handling, 595, 601
- moving multiple files, 178
- moving multiple folders, 166
- not using when copying, moving or deleting individual files, 178
- not using when copying, moving or deleting individual folders, 167
- PowerShell comparison operators for, 81
- using asterisk (\*) symbol as, 568
- using PowerShell -like operator to check for, 718

Win32

- LDAP error codes for ADSI 2.0, 807–810
- LDAP error codes with, 804–807

Win32\_Service WMI object, 542–543, 548

Win32-Account object, 552

- Win32-Process object, 551
- Win32-Session object, 551
- window styles, programs and scripts, 123–124
- Windows Explorer, 92–93
- Windows file time, 567–568
- Windows Internet Name Service (WINS), managing, 237–239
- Windows Registry, 231–242
  - creating new keys, 235
  - deleting keys and values, 236
  - overview of, 231–232
  - reading keys and values, 233–234
  - reconfiguring network services through, 236–242
  - understanding environment variables, 117
  - understanding structure, 232–233
  - working with file system vs., 537–539
  - writing keys and values, 234–235
- Windows Script Host. *See* WSH (Windows Script Host)
- Windows scripting, 3–12
  - overview of, 3–4
  - Windows PowerShell architecture, 10–12
  - WSH architecture. *See* WSH (Windows Script Host)
- Windows Scripting API, 735–747
  - drives collection, 737–738
  - getResource static method, 737
  - printers collection, 738
  - Script.Signer object, 741
  - StdErr stream, 739
  - StdIn stream, 738–739
  - StdOut stream, 739
  - WScript object, 741–742
  - WshArguments collection, 740
  - WshController object, 742
  - WshEnvironment object, 742–743
  - WshNamed collection, 740
  - WshNetwork object, 743
  - WshRemote object, 744
  - WshRemoteError object, 744
  - WshScriptExec object, 745
  - WshShell object, 745–746
  - WshShortcut object, 746
  - WshSpecialFolders object, 747
  - WshUnnamed collection, 740–741
  - WshUrlShortcut object, 747
  - XML elements, 735–737
- Windows Server 2008 Administrator's Pocket Consultant*, 301
- Windows services. *See* services
- WinNT ADSI provider
  - core object model, 332–333
  - domain account policies, 347–353
  - group accounts, 372–376
  - local computer properties, 353–356
  - obtaining shared folder objects through, 415–416
  - overview of, 309–312
  - user accounts. *See* user accounts
- WINS (Windows Internet Name Service), managing, 237–239
- WMI (Windows Management Instrumentation), 581–604
  - accessing services through, 542–543
  - discovering objects, 596
  - displaying information, 583–586
  - Hyper-V case study for, 600–602
  - object methods, 599
  - object properties, 596
  - overview of, 581–583
  - queries, choosing how to write, 587–588
  - queries, finding WMI objects by association, 588–591
  - queries, type accelerators for, 592–594
  - queries across computers, 594–596
  - updating objects, 596
- WMI Query Language (WQL), 587, 588–591
  - [WMICLASS], 593–594
- working directory, setting link shortcut, 216, 219
- workstations
  - using NET CONFIG WORKSTATION, 831
  - using NET STATISTICS, 838–839
  - using NSLOOKUP, 845–846
- WQL (WMI Query Language), 587, 588–591
- wrap switch, format-table, 503
- Write method, 194
- WriteBlankLines method, 194–195
- WriteChar function, 635
- Write-error cmdlet, 496
- WriteEvent function, 638
- Write-host cmdlet, 491–492
- WriteLine function, 635–636
- WriteLine method, 194–195
- Write-Progress cmdlet, 501–503
- Write-Verbose cmdlet, 493–494
- writing
  - events to application event logs, 245–247
  - to file, 192–195
  - registry keys and values, 234–235
  - write command in PowerShell, 493
- WScript
  - defined, 5
  - examining information, 110–111
  - methods and properties, 107–108
  - running scripts with, 5, 91–95
- WScript object, 147–152, 741–742
- WScript.Shell, 121–124
- WScript.WshNetwork object, 108
- WScript.WshShell object, 108–109, 118–120
- .wsf files, 8, 91, 96
- WSH (Windows Script Host), 4–10
  - core object model, 6–8
  - creating scripts. *See* scripts
  - getting started with, 4–5
  - JScript objects for Windows scripting, 10
  - scripting basics. *See* scripts
  - scripting engines, 8–9
  - scripting hosts provided by, 8, 91–92

- using and running scripts, 5–6
- VBScript objects for Windows scripting, 9
- Windows Scripting API. *See* Windows Scripting API
- WSH ActiveX control, 5
- .wsh files, 8
- WshArguments collection, 115–116, 740
- WshController object, 124–126, 742
- WshEnvironment object, 119–120, 742–743
- WshNamed collection, 740
- WshNetwork object, 743
- WSH.ocx ActiveX control, 6
- WshRemote object, 744
- WshRemoteError object, 744
- WshScriptExec object, 745
- WshShell object, 745–746
- WshShortcut object
  - defined, 214

- properties of, 216
- reference guide, 746
- WshSpecialFolders object, 747
- WshUnnamed collection, 740–741
- WshUrlShortcut object, 214, 747

## X

- XML
  - elements, reference guide, 735–737
  - formatting objects in PowerShell, 509–512
  - handling in PowerShell, 715–718
  - markup tags. *See* batch scripts
  - processing in PowerShell, 518–520
- <?XML ?> element, 103–104, 735
- XOr comparisons, UserFlags, 364
- x/y coordinates, input boxes, 132