

Index

SYMBOLS & NUMERICS

~directory, as shortcut, 24
 /etc/services file, 51
 ~modifier, 35
 \$^O (\$OSNAME) scalar, 184
 \$server@aget_last_error() method, 413
 #! (shebang), 153, 184
 /tmp folder, 115
 \$vm@raget_last_error() method, 413–414
 64-bit platform, 353
 902 port, verifying availability of, 50

A

accessing host settings, 62
 adapter architecture, selecting, for virtual machine, 105–106
 Add Reference screen (Visual Studio Team System), 331
 Add Solution to Source Control screen (Visual Studio Team System), 320, 321
 Add Team Foundation Server dialog box (Visual Studio Team System), 313
 Add to Source Control dialog box (Visual Studio Team System), 346
 Add to SourceSafe dialog box (Visual SourceSafe), 286–287
 adding
 build distribution node, 298–299
 development project
 to source control, 285–288
 to team project, 318–322
 directory to Additional Include Directories window, 218
 source control, 289–291
 trigger, 288–289
 Additional Dependencies option, 219, 220
 Additional Include Directories option, 216
 Additional Include Directories window, 217, 218

Add/Remove Team Foundation Server screen (Visual Studio Team System), 313, 314
 Administering permission, 93
 Advanced button (settings pane), 143
 Advanced dialog box, 379, 380
 Advanced section (Options tab), 82–84
 Advanced tab (System Properties screen), 210, 211
 Allocate all disk space now option (New Virtual Machine Wizard), 363
 allowing File and Printer sharing, 401
 AMD
 PAE standard and, 13
 processor requirements, 12
 answer file
 creating, 369–372
 storing, 121
 answers, default, to install queries, 26, 36, 45
 antivirus software, 134
 API (application program interface). *See also* Programming API; scripting APIs; VmCOM API; VmPerl API
 automation and, 151
 installing, 42
 supported platforms, 153–154
 tasks for automation, 152
 Application Settings screen (Win32 Application Wizard), 214
 Applications® System Tools, 33, 52
 architecture, hosted versus hypervisor, 3–5
 archiving
 base images, 140
 development environment, 127, 131
 Sysprep folder, 372
 Ask me (Snapshots section), 77
 asynchronous operations, 226–230
 authenticating and logging in via code, 262–263
 Autofit section (Display tab), 72
 Automatic Bridging tab (Virtual Network Editor), 86
 automation, tasks for, 152. *See also* API

B

bandwidth, 14

base image. See also development image; testing image

- archiving, 140
- creating, 98–109
- description of, 97, 128, 138
- freezing Windows guest, 116–123
- installing operating system onto, 109–113
- ISO image library, creating, 144–148
- library of, organizing, 137–140
- licensing issues, 98, 123
- naming, 137–138
- preparing for use, 140–141

batch file

- placing in team project, 342–347
- in VMware Tools, 281

binaries

- copying to virtual machine, 337–338
- obtaining prior to installation, 19, 24
- pushing up to desktop, 399–400

BIOS, virtual, 74

blocking until job is complete, 228–230

bridged networking, 28–29

Browsing permission, 93

build

- configuring, 296–299
- forcing, 388–391

build distribution node, adding, 298–299

Build Explorer (Visual Studio Team System), 322

build integration, configuring

- final scripts, 403–407
- overview of, 380
- project setup
 - CCNet Tray application, installing, 382–383
 - ccnet.config file, 383–394
 - CruiseControl.NET, installing, 381–382
 - VMware Server integration, 394–402

build log window (Visual Studio Team System), 329

build publisher, configuring, 394

Build Servers section (CruiseControl.NET Tray Settings screen), 292

build task, coding, 333–337

Build Team Build Project screen (Visual Studio Team System), 329

BusLogic adapter, 105

C

C compiler and building modules, 48

C#, scripting VMware Tools with, 283–284

.cab file, 117, 368

callback function

- asynchronous operations, job handle, and, 230
- VixHost_FindItems() function and, 240–243

CCNet (CruiseControl.NET). See also CCNet Tray application; ccnet.config file

- Additional Configuration screen, 382
- build, configuring, 296–299
- build distribution log, adding, 298–299
- configuring, 285–299, 383–394

- continuous integration and, 278

- Dashboard, 295–296, 391

- description of, 151, 285

- developmental project, adding to source control, 285–288

- error handling, 302

- installing, 381–382

- IP address of virtual machine, retrieving, 305

- main screen, 291, 294

- parsing list of virtual machines, 302–304

- powering on first available machine, 304

- retrieving list of registered virtual machines, 300–302

- source control, adding, 289–291

- trigger, adding, 288–289

- version of, 298

- VMware Server integration, 299–311

- XML logging, 298

CCNet Dashboard, 295–296

CCNet icon, 291

CCNet Tray application

- configuring, 291–295

- forcing build, 388–391

- installing, 382–383

- main interface, 388, 389, 390

- Project dialog box, 390

- Settings screen, 388, 389

CCNet Tray icon, 388

ccnet.config file

- build publisher, 394

- calling Perl script after build is completed, 310–311

- categories of, 288

- code, 310–311, 403–404

- external script for VMware Server integration, 394

- labeler, 385–386

- overview of, 384

- root project node, 384, 385

- source control, 387

- state file location, 384–385

- trigger block, 387–388

CCNetLabel variable, 306

CD-ROM autorun, disabling, 20, 22

CD-ROM icon, 148

chip set, virtual, 74

code

- authenticating and logging in via, 262–263

- ccnet.config file, 310–311, 403–404

- example1.pl, 188–190

- example1.vbs, 163–164

- example1.wsf, 163

- example2.pl, 196–199

- example2.vbs, 169–170

- example2.wsf, 169

- example3.pl, 200–203

- example3.vbs, 172–173

- example3.wsf, 172

- example4.vbs, 175–176

- example4.wsf, 174

- getIP.vbs file, 280

- getIP.wsf file, 280

- Program.cs file, 283

- Program.vb file, 284

- test-winxppro-sp2 file, 302–303

- VMIntegration class, 338–342
- vmSetIP.cs file, 375–376
- vmTestBank.pl file, 307–310, 404–407
- writing from within development virtual machine, 9
- coding build task, 333–337**
- collaborating on development image, 133**
- COM (Component Object Model), 153, 155**
- commands**
 - Create Launcher, 52–53
 - mount, 114
 - shutdown, 115
 - su -, 27
- Commands area (main pane)**
 - components of, 60
 - Edit virtual machine settings, 72
- compiling against Vix with Visual C++**
 - include statement and, 215
 - overview of, 209–210
 - pointing project
 - to header files, 216–218
 - to libraries used, 218–220
 - setting up project, 210–215
- compiling kernel modules, 48–50**
- Completing Setup Manager screen (Setup Manager Wizard), 371–372**
- Component Object Model (COM), 153, 155**
- compressing base image for archiving, 140**
- compression utility, 25, 140**
- configuration file for virtual development machine, renaming, 365–367**
- Configure Host option (Home tab), 62**
- Configure Product screen, 22**
- configuring. See also ccnet.config file**
 - build, 296–299
 - CCNet Tray application, 291–295
 - CruiseControl.NET (CCNet), 285–299, 383–394
 - Linux installation, 27–34
 - MSBuild, 296–299, 391–393
 - team build, 322–328
 - testing images as nonpersistent, 378–380
 - user access levels, 93–96
- Configuring permission, 93**
- Confirm Team Project Settings screen (New Team Project wizard), 317**
- Connect to Team Foundation Server screen (Visual Studio Team System), 314**
- connecting to server**
 - function example, 271–275
 - Perl script, 396
 - VixHost_Connect function and, 237–239
 - VmCtl object and, 169–171
- Connection tab (host settings), 66, 67**
- connection to server, verifying, 187, 192**
- console preferences (server console)**
 - Display tab, 71–72
 - Hot Keys tab, 70–71
 - Input tab, 68–69
 - overview of, 62, 66–67
 - Workspace tab, 68
- constants, VmPerl API, 204–207**
- context menu (Linux), 52–53**
- continuous integration, 278, 349**
- Continuum continuous integration package, 349**
- copy and paste section (Input tab), 69**
- copying**
 - base image, 134
 - binaries to virtual machine, 337–338
 - file
 - from guest machine to host machine, 266–267
 - from host machine to guest machine, 265–266
 - to virtual machine, 271–275
- C:\Program Files\VMware\VMware VIX directory, 210
- Create Launcher command (Linux), 52–53**
- creating**
 - answer file, 369–372
 - base image, 133–134
 - custom assembly for integration, 330–333
 - ISO image library, 144–148
 - snapshot, 251–252
 - team project, 312–318
 - temporary object, 236
 - virtual disk, 360
 - virtual machine, 98–109
 - vmware-users group, 95–96
- cross-platform Perl scripts, 183–185**
- CruiseControl.NET (CCNet). See also CCNet Tray application; ccnet.config file**
 - Additional Configuration screen, 382
 - build, configuring, 296–299
 - build distribution log, adding, 298–299
 - configuring, 285–299, 383–394
 - continuous integration and, 278
 - Dashboard, 295–296, 391
 - description of, 151, 285
 - developmental project, adding to source control, 285–288
 - error handling, 302
 - installing, 381–382
 - IP address of virtual machine, retrieving, 305
 - main screen, 291, 294
 - parsing list of virtual machines, 302–304
 - powering on first available machine, 304
 - retrieving list of registered virtual machines, 300–302
 - source control, adding, 289–291
 - trigger, adding, 288–289
 - version of, 298
 - VMware Server integration, 299–311
 - XML logging, 298
- CruiseControl.NET Tray Settings screen, 292, 294**
- Ctrl+Alt+Del, 70**
- Ctrl+Alt+Enter, 71**
- Ctrl+Alt+F1 to F6, 114**
- Ctrl+Alt+Ins, 70**
- Ctrl+Alt+Space, 114**
- Ctrl+Shift+Alt+Ins, 70**
- cursor behavior, 69**
- custom assembly, creating for integration, 330–333**
- Custom Setup screen, 20, 21**
- Customer Information screen, 22**

D

- debugging mode, 84**
- default answers to install queries, 26, 36, 45**
- default hardware items and settings for operating system, 98–99**
- default hot key, changing, 70**
- default labeller, 385**
- default port, 31**
- defining**
 - goals, 353–354
 - target environment, 351–353
- deleting**
 - virtual development machine floppy drive, 365
 - virtual machine, 250–251
- deploy.cab file, 116, 117**
- deploying development image, 135**
- deployment, setting up virtual development machine software for, 367–374**
- depreciation of scripting APIs, 179, 180**
- desktop of virtual machine, pushing project payload to, 305–306**
- determining**
 - number of root snapshots, 253–254
 - operational status, 204
 - results of pending job handle, 227–230
- developer machine, 8**
- development environment**
 - archive of, 127, 131
 - creating base images for, 131–135
 - integrating power of virtual machines into, 277
 - overview of, 125
 - recovery of, 126–127
 - virtual machines and, 126–127
- development image**
 - collaborating on, 133
 - creating, 131–135
 - deploying, 135
 - description of, 138
 - planning, 132–133
 - setting up, 354–364
 - testing, 134–135
- development machine, virtual**
 - deleting floppy drive, 365
 - renaming configuration file, 365–367
 - setting up, 355–364
 - setting up software for deployment, 367–374
- development process, managing, 353**
- development server, virtualization of, 9**
- developmental project, adding**
 - to source control, 285–288
 - to team project, 318–322
- Device status screen, 110**
- Devices area (main pane), 60**
- Devices tab (host settings), 66**
- DHCP Settings window, 88–89**
- DHCP tab (Virtual Network Editor), 92**
- directory**
 - adding to Additional Include Directories window, 218
 - shortcut (~), 24
 - working, 298

- Disable acceleration option (Advanced section), 84**
- Disable memory page trimming option (Advanced section), 84**
- disabling. See also enabling**
 - CD-ROM autorun, 20, 22
 - memory page trimming, 84
 - simple file sharing, 400
- disconnecting from server, 239–240. See also connecting to server**
- disk**
 - independent
 - platform testing with, 130
 - snapshots and, 144
 - types of, 143
 - nonpersistent
 - description of, 143
 - platform testing with, 130
 - using, 143–144
 - persistent
 - description of, 143, 378
 - platform testing with, 130
 - physical
 - installing operating system from
 - setting up, 361
 - requirements for, 13–14
 - virtual
 - creating, 360
 - pre-allocating, 13, 364
 - size of, 363
 - storing, 108–109
- disk size**
 - compression of files and, 140
 - configuring for virtual machine, 107–108
- Display tab (console preferences), 71–72**
- displaying state of registered virtual machines, 171–173**
- documentation, 127**
- downloading installation package, 19, 24–25**
- drag and drop functionality, 69**

E

- Edit System Variable screen, 212**
- Edit virtual machine settings (Command section), 72**
- enabling. See also disabling**
 - automatic startup and shutdown of guest, 81–82, 83
 - File and Printer sharing, 401
 - MUI and console ports in firewall, 54–55
 - periodic logging, 84
- End User License Agreement, 98**
- enforcing standards, 133**
- enumeration**
 - definition of, 160
 - VmCOM API, 177–180
- environment. See also development environment**
 - defining target, 351–353
 - rollout of new, 127
 - testing, isolating, 91
 - X, restarting, 115
- Environment Variables button, 210**
- Environment Variables screen, 211**

error codes

- Programming API, 225–226, 417–421
- VmCOM API, 409–411
- VmPerl API, 413–415

error handling, 302

- /etc/services file, 51

Ethernet card, virtual, 75

evaluation, built-in functions for, 225–226

event types and callback function, 230

examples

- of functions, 230
- integration scenarios, 209
- VixCopyToGuest.cpp file, 271–275
- VixRegisterVM.cpp file, 270–271
- VmCollection object, 171–173
- VmCtl object, 168–171
- VmQuestion object, 174–177
- VmServerCtl object, 163–164
- VMware::VmPerl::Question module, 200–204
- VMware::VmPerl::Server module, 188–191
- VMware::VmPerl::VM module, 195–199

Exceptions tab (Windows Firewall dialog box), 401–402

Execute() method, 333, 334–335

executing program on guest machine, 263–265

exit, in Perl, 191

expanding virtual disk, 13

F

FAT16-based file system, 105

FAT32-based file system, 105

file. See also ccnet.config file

- answer
 - creating, 369–372
 - storing, 121
- batch, 281, 342–347
- .cab, 117, 368
- copying
 - from guest machine to host machine, 266–267
 - from host machine to guest machine, 265–266
 - to virtual machine, 271–275
- deploy.cab, 116, 117
- /etc/services, 51
- header, pointing project to, 216–218
- library, pointing project to, 218–220
- .log, 101
- nvram, 101
- .vbs, 160
- .vmdk, 101
- .vmdkREDO, 101
- .vmsn, 101
- .vmx, 93, 101
- .vmx.sav, 101
- .wsf, 160
- XML, 333, 347–348

File and Printer sharing, 305, 401

File Download screen, 39

file extensions, 101

File® New® Project, 212

File® Source Control, 286, 319–320

Filter trigger (CCNet), 289

finding running virtual machine, 240–243

firewall, enabling MUI and console ports in, 54–55

flexibility

- of image names, 139
- of VMware products, 8

floppy drive, virtual, 74, 365

Folder Options dialog box, View tab, 401

Force Build button (CCNet), 295

forcing

- build, 388–391
- vmExecutionState_Stuck state, 174–177
- VM_EXECUTION_STATE_STUCK state, 200–204

freeing string buffer, 235–236

freezing Windows guest, 116–123

full screen view, 72

functions (Programming API)

- built-in, for evaluation, 225–226
- error code and, 225–226
- general use, 230–237
- host machine operations, 237–244
- job handle operations, 268–269
- referencing with include statement, 215
- virtual machine operations
 - VixVM_CopyFileFromGuestToHost function, 266–267
 - VixVM_CopyFileFromHostToGuest function, 265–266
 - VixVM_CreateSnapshot function, 251–253
 - VixVM_Delete function, 250–251
 - VixVM_GetNumRootSnapshots function, 253–254
 - VixVM_GetRootSnapshot function, 253–254
 - VixVM_InstallTools function, 258–259
 - VixVM_LoginInGuest function, 262–263
 - VixVM_Open function, 245–246
 - VixVM_PowerOff function, 247–248
 - VixVM_PowerOn function, 246–247
 - VixVM_RemoveSnapshot function, 255–256
 - VixVM_Reset function, 248–249
 - VixVM_RevertToSnapshot function, 256–258
 - VixVM_RunProgramInGuest function, 263–265
 - VixVM_Suspend function, 248–249
 - VixVM_UpgradeVirtualHardware function, 261–262
 - VixVM_WaitForToolsInGuest function, 259–261

G

General section (Options tab), 75–76

General tab (host settings), 62–63

get, definition of, 160

GetProperties() function, 224

get_tools_last_active() method (VmPerl API), 195

goals, defining, 353–354

graphics, virtual, 74

guest. See also base image

- copying file from, to host machine, 266–267
- copying file to, from host machine, 265–266
- description of, 6
- enabling automatic startup and shutdown of, 81–82, 83
- executing program on, 263–265
- freezing with Sysprep, 116–123

guest (continued)

- mouse, keyboard, and, 68–69
- per processor, recommended standard for, 12

GZip (GNU zip) compression utility, 25, 140

H

handle properties (Programming API), 222–224

handle types (Programming API), 222

handles (Programming API)

- description of, 220–222
- representing root snapshot, getting, 254–255

hard disk, requirements for, 13–14

hardware

- default, and settings for operating system, 98–99
- for remote client, 17–18
- virtual machine, specifications for, 73–75

hardware icons, 112

Hardware tab (Virtual Machine Settings), 109–110, 148

header file, pointing project to, 216–218

Help® Enter Serial Number, 32

Home tab (server console), 61–62, 355

host handle, 222

host hardware requirements, 11–14

host machine

- copying file from guest machine to, 266–267
- copying file to guest machine from, 265–266
- description of, 6

host machine operations (Programming API), 237–244

host operating system

- description of, 6, 24
- requirements for
 - Linux, 15–17
 - overview of, 14
 - Windows, 15

host settings (server console)

- Connection tab, 66, 67
- Devices tab, 66
- General tab, 62–63
- Memory tab, 63–64
- Priority tab, 64–65

Host Virtual Adapters tab (Virtual Network Editor), 91

Host Virtual Network Mapping tab (Virtual Network Editor), 87–90

Host® Virtual Network Settings, 84

hosted architecture, 3–5

host-only networking, 30

hot key, default, changing, 70

Hot Keys tab (console preferences), 70–71

HTTP session time out, 36

hypervisor architecture, 3–5

I

icons

- CCNet, 291
- CCNet Tray, 388
- CD-ROM, 148
- hardware, 112
- plus, 320
- VMware Tools, 116, 376

IDE drive, virtual, 74

identifying

- type of handle provided as parameter, 233–234
- type of property stored in handle, 232–233

image. See base image; development image; ISO image; testing image

imaging workstation, 127

include statement, 215

independent disk

- platform testing with, 130
- snapshots and, 144
- types of, 143

Infotype constant, 205–206

Input grabbed state, 65

Input tab (console preferences), 68–69

Input ungrabbed state, 65

installation

- of API, 42
- of CCNet Tray application, 291, 382–383
- of CruiseControl.NET, 285, 381–382
- host hardware requirements for, 11–14
- host operating system requirements for, 14–17
- of Internet services daemon, 50–51
- on Linux, 24–34
- of operating system onto base image, 109–113
- overview of, 34
- of remote client
 - on Linux, 43–48
 - on Windows, 38–42
- remote client requirements for, 17–18
- troubleshooting, 48–55
- VMware products and, 18
- of VMware Server Management Interface on Linux, 34–38
- of VMware Tools
 - on Linux guests, 113–116
 - VixVM_InstallTools function, 258–259
 - on Windows guests, 113
- on Windows, 19–24
- of xinetd, 50–51

integration. See also CruiseControl.NET; Visual Studio Team System

- configuring external script for, 394
- continuous, 278, 349
- creating custom assembly for, 330–333
- Linux and, 349
- overview of, 277
- Perl script for, 394–402
- source control and, 353

Intel

- PAE standard and, 13
- processor requirements, 12

Interacting permission, 93

Internet Explorer-Security Warning screen, 40

Internet services daemon, installing, 50–51

Interval trigger (CCNet), 289, 388

Inventory pane (server console), 58–59

IP address

- grabbing, 398–399
- of guest, retrieving, 375–377
- of virtual machine, retrieving, 305

ISO image

- creating, 145–146
- description of, 144
- installing operating system from, 111
- library, creating, 144–148
- using, 146–148

isolating testing environment, 91

`ISupportErrorInfo` interface, 409–410

iterating list of registered virtual machines, 171–173

iteration labeller, 385

J**job handle**

- asynchronous operations and, 227–230
- description of, 222
- properties, 223–224

job handle operations (Programming API), 268–269

Just power off (Snapshots section), 77

K**kernel**

- compiling modules for, 48–50
- installing modules into, 27–28
- upgrading, 34

keyboard

- focus of, 69
- virtual, 75

L

labeler, configuring, 385–386

labeler, default, 385

launching VMware Server Console manually, 52–53

`libdb.so.3`, troubleshooting missing, 51–52

library

- of base images, 6, 137–140
- of ISO images, creating, 144–148
- Perl, 184
- placing in team project, 342–347
- `ThoughtWorks.CruiseControl.MSBuild`
- `.XMLLogger` library, 392

library file, pointing project to, 218–220

License Agreement screen

- Setup Manager Wizard, 120, 370
- VMware Server, 20

licensing issues, 98, 123

Linux

- creating ISO files in, 147
- host operating system requirements, 15–17
- platform testing, 129
- remote client
 - installing on, 43–48
 - requirements for, 18
- user access, configuring on, 95–96
- virtual networking, configuring on, 85
- VMware Server, installing on, 24–34
- VMware Server integration under, 349
- VMware Server Management Interface, installing, 34–38
- VMware Tools, installing on, 113–116
- VMware Tools scripting in, 279

local system account, running virtual machine under, 103

localhost server, as default, 23

location

- for base image, choosing, 99
- for storing virtual disk, choosing, 108–109
- `.log` file, 101

Log virtual machine progress periodically option (Advanced section), 84

logging in via code, 262–263

LSI adapter, 105

M

main pane (server console), 59–61

management interface. See VMware Server Management Interface, installing; web management interface

masking error code, 225–226

memory

- requirements for, 12
- virtual, 74
- for virtual machine, 104–105, 360

Memory for the Virtual Machine screen (New Virtual Machine Wizard), 104, 360

memory page trimming, disabling, 84

Memory tab (host settings), 63–64

methods

- definition of, 160
- `VmCtl` object, 166–168
- `VmServerCtl` object, 162–163
- `VMware::VmPerl::ConnectParams` module, 186
- `VMware::VmPerl::Question` module, 199
- `VMware::VmPerl::Server` module, 187–188
- `VMware::VmPerl::VM` module, 191–195

Microsoft .NET Framework, 283, 297

mode

- debugging, 84
- quick switch, 72

modifier (~), 35

modules

- compiling, 48–50
- installing into kernel, 27–28

mount command, 114

mouse

- focus of, 68–69
- virtual, 75

moving virtual machine from one server to another, 8

MSBuild, configuring, 296–299, 391–393

multiplatform testing, 9

My Computer® Properties® Advanced, 188

My Computer® Tools® Folder Options, 400

N

Name and Organization screen (Setup Manager Wizard), 371

Name the Virtual Machine screen (New Virtual Machine Wizard), 100, 358

naming. See also renaming

- base images, 99, 137–138
- production images, 140–141
- projects, 354–355

naming (continued)

- test images, 303–304
- virtual machines, 138
- NAT (Network Address Translation) networking option, 29–30**
- NAT network, physical IP address and, 89**
- NAT Settings window, 90**
- NAT tab (Virtual Network Editor), 92**
- .NET Framework v2.0, 283, 297**
- Netbios timeout settings (Host Virtual Network Mapping tab), 90**
- Network Address Translation (NAT) networking option, 29–30**
- Network Connections dialog box, 402**
- network controller, requirements for, 14**
- Network Type screen (New Virtual Machine Wizard), 105, 361**
- networking**
 - bridged, 28–29
 - host-only, 30
 - NAT, 29–30
 - virtual
 - Automatic Bridging tab, 86
 - Host Virtual Network Mapping tab, 87–90
 - overview of, 84–85, 360–361
 - Summary tab, 85
- networking configuration for Linux, 28–30**
- New or Existing Answer File screen (Setup Manager Wizard), 119, 369**
- New Project dialog box, 213**
- New Team Build Type Creation wizard (Visual Studio Team System)**
 - Select and order solutions to build screen, 324
 - Select build location screen, 326
 - Select build options screen, 327
 - Select configurations to build screen, 325
 - Welcome screen, 323
- New Team Project wizard (Visual Studio Team System)**
 - description of, 315
 - Specify the Settings for the Project Portal screen, 317
 - Specify the Source Control Settings screen, 317
 - Specify the Team Project Settings screen, 316
- New Virtual Machine option (Home tab), 61**
- New Virtual Machine Wizard**
 - Allocate all disk space now option, 363
 - Memory for the Virtual Machine screen, 104, 360
 - Name the Virtual Machine screen, 100, 358
 - Network Type screen, 105, 361
 - overview of, 98
 - Processor Configuration screen, 103–104, 359
 - Select a Disk screen, 106, 362
 - Select a Disk Type screen, 107, 362
 - Select a Guest Operating System screen, 100, 357
 - Select I/O Adapter Types screen, 106, 361
 - Select the Appropriate Configuration screen, 99, 356
 - Set Access Rights screen, 102, 358
 - Specify Disk Capacity screen, 107–108
 - Specify Disk File screen, 108–109, 364
 - Startup/Shutdown options, 357–358
 - Startup/Shutdown Options screen, 102–103, 359
 - Welcome screen, 356
- 902 port, verifying availability of, 50**
- nonpersistent, configuring testing images as, 378–380**

nonpersistent disk

- description of, 143
 - platform testing with, 130
 - using, 143–144
- Notes area (main pane), 60**
- nvram file, 101**

O

object

- definition of, 159
 - temporary, creating, 236
- object-oriented programming (OOP), 159–160**
- Open dialog box, 282, 377**
- Open Existing Virtual Machine option (Home tab), 61**
- Open Virtual Machine screen, 142, 367**
- opening. See also starting**
- .cab file, 368
 - ccnet.config file, 288, 384
 - New Virtual Machine Wizard, 98
 - virtual machine, 245–246
 - Virtual Network Settings window, 84
 - Visual Studio IDE, 312
 - web management interface, 37
- operating system. See also Linux; Windows**
- abbreviations for, 139
 - default hardware items and settings for, 98–99
 - host, 6, 14–17, 24
 - installing onto base image, 109–113
- operational status, determining, 204**
- Options tab**
- Virtual Machine Settings window
 - Advanced section, 82–84
 - General section, 75–76
 - Permissions section, 78, 79
 - Power section, 76–77
 - sections in, 141–142
 - Snapshots section, 77–78
 - Startup/Shutdown section, 78–82
 - web management interface, 81
- organizing library of base images, 137–140**
- \$OSNAME (\$^O) scalar, 184**
- overkill, 354**

P

- PAE (Physical Address Extension), 13**
- parallel (LPT) port, virtual, 75**
- parsing list of virtual machines, 302–304**
- patch level, 139**
- PCI slot, virtual, 74**
- Pending Changes tab (Visual Studio IDE), 320, 321, 347**
- Pending Checkins page (Visual SourceSafe), 287**
- performance and ISO images, 144**
- periodic logging, enabling, 84**
- Perl**
 - cross-platform scripts, 183–185
 - description of, 153, 183
 - non-zero returns and, 413
 - regular expressions, 303–304
 - VMware Server integration scripts, 394–402
- Perl interpreter, 154, 183**

- permissions, 93, 96**
 - Permissions section (Options tab), 78, 79**
 - persistent disk**
 - description of, 143, 378
 - platform testing with, 130
 - Physical Address Extension (PAE), 13**
 - physical disk**
 - installing operating system from, 111
 - setting up, 361
 - planning base image, 132–133**
 - platform. See also platform testing**
 - multiple, available on single machine, 7
 - 64-bit, 353
 - supported, 153–154, 352
 - target environment and supported, 352
 - platform availability for support analysis, 131**
 - platform testing**
 - Linux, 129
 - multiplatform testing, 9
 - overview of, 128
 - with persistent and nonpersistent disks, 130
 - with snapshots, 129
 - plus icon, 320**
 - polling**
 - guest machine to determine if VMware Tools is running, 259–261
 - for job completion, 227–228, 269
 - port**
 - default, 31
 - in firewall, enabling, 54–55
 - verifying availability of 902, 50
 - virtual, types of, 75
 - Port Forwarding button (Host Virtual Network Mapping tab), 90**
 - post-release maintenance, virtual machines for, 131**
 - Power section (Options tab), 76–77**
 - power transition, behavior during, 204–205**
 - powering off virtual machine, 247–248**
 - powering on**
 - first available machine, 304
 - function for, 246–247, 271–275
 - virtual machine, 111, 169–171
 - pre-allocated virtual disk, 13, 364**
 - preparing base images for use, 140–141**
 - Priority tab (host settings), 64–65**
 - process control for target environment, 353**
 - processor**
 - product activation and, 127
 - requirements for, 12
 - virtual, 73
 - Processor Configuration screen (New Virtual Machine Wizard), 103–104, 359**
 - product activation and processor, 127**
 - Programming API (Vix)**
 - asynchronous operations and job handle, 226–230
 - bug in v1.0 of, 275
 - compiling against, with Visual C++, 209–220
 - description of, 154, 155, 156, 209
 - error codes, 225–226, 417–421
 - examples, 269–275
 - functionality of, 153
 - functions, 230–237
 - handle properties, 222–224
 - handle types, 222
 - handles, 220–222
 - host machine operations, 237–244
 - job handle operations, 268–269
 - release notes for version 1.0, 209
 - as supported scripting method, 207
 - as thread safe, 227
 - virtual machine operations
 - VixVM_CopyFileFromGuestToHost function, 266–267
 - VixVM_CopyFileFromHostToGuest function, 265–266
 - VixVM_CreateSnapshot function, 251–253
 - VixVM_Delete function, 250–251
 - VixVM_GetNumRootSnapshots function, 253–254
 - VixVM_GetRootSnapshot function, 253–254
 - VixVM_InstallTools function, 258–259
 - VixVM_LoginInGuest function, 262–263
 - VixVM_Open function, 245–246
 - VixVM_PowerOff function, 247–248
 - VixVM_PowerOn function, 246–247
 - VixVM_RemoveSnapshot function, 255–256
 - VixVM_Reset function, 248–249
 - VixVM_RevertToSnapshot function, 256–258
 - VixVM_RunProgramInGuest function, 263–265
 - VixVM_Suspend function, 248–249
 - VixVM_UpgradeVirtualHardware function, 261–262
 - VixVM_WaitForToolsInGuest function, 259–261
 - vmrun utility and, 157
 - Programs® Settings® Session® Startup Programs, 116**
 - Project dialog box (CCNet Tray), 390**
 - Project screen (CCNet), 293**
 - properties**
 - definition of, 160
 - handle, 222–224
 - Perl and, 185
 - VmCollection object, 171
 - VmConnectParams object, 161–162
 - VmCtl object, 164–166
 - VmQuestion object, 174
 - VmServerCtl object, 162
 - Property Pages screen**
 - C/C++ node, 216, 217
 - Configuration Properties node, 216
 - Linker node, Input option, 218–219
 - ProVMwareSampleApp screen (Visual Studio Team System), 344**
 - pushing project payload to desktop of virtual machine, 305–306**
- ## Q
- quick switch mode, 72**
- ## R
- read and execute permissions, 93**
 - read permissions, 93**
 - read, write, and execute permissions, 93, 96**
 - recycling testing image, 143–144**

reference count for handle, 221, 231

References tab (Solution Explorer, VSTS), 332

referencing

functions with `include` statement, 215

objects with handles, 220

registering virtual machine

remotely, 270–271

with server, 163–164, 188–191, 243

regular expressions (Perl), 303–304

Release screen (Visual Studio Team System), 343

Release To Manufacture (RTM) version, 128, 139

remote client

definition of, 17, 38

installing

on Linux, 43–48

on Windows, 38–42

requirements for, 17–18

remote connection for server console, configuring, 31

removing snapshot, 255–256

renaming

virtual development machine configuration file, 365–367

virtual disk file, 108, 141, 363

Reseal method (Sysprep), 122, 374

resetting virtual machine, 248–249

restarting X environment, 115

retrieving

IP address

of guest, 375–377

of virtual machine, 305

list of registered virtual machines, 300–302

property from handle, 224, 231–232

string description of error passed in, 234–235

Revert to snapshot (Snapshots section), 77

reverting

to snapshot, 129

to state present in snapshot, 256–258

rollout of new environment, 127

root project node, configuring, 384, 385

root snapshot, determining number of, 253–254

RTM (Release To Manufacture) version, 128, 139

Run with debugging information option (Advanced section), 84

running

base image with VMware Player, 135

power scripts, 76

running virtual machine, finding, 240–243

S

saving script, 188

scalar, 184

Schedule trigger (CCNet), 289, 388

script, saving, 188

scripting, 151. See also VMware Tools scripting

scripting APIs

depreciation of, 179, 180

future of, 207

supported platforms, 153–154

Scripts tab (VMware Tools Properties screen), 281, 282, 376, 377

SCSI device, virtual, 74

security identifier (SID), 116, 142

Security Level Configuration screen, 54, 55

Security tab (Virtual Machine Properties), 94

security, user access level options, 92–93

Select a Disk screen (New Virtual Machine Wizard), 106, 362

Select a Disk Type screen (New Virtual Machine Wizard), 107, 362

Select a Guest Operating System screen (New Virtual Machine Wizard), 100, 357

Select and order solutions to build screen (New Team Build Type Creation wizard), 324

Select build location screen (New Team Build Type Creation wizard), 326

Select build options screen (New Team Build Type Creation wizard), 327

Select configurations to build screen (New Team Build Type Creation wizard), 325

Select I/O Adapter Types screen (New Virtual Machine Wizard), 106, 361

Select the Appropriate Configuration screen (New Virtual Machine Wizard), 99, 356

serial (COM) port, virtual, 75

serial number, 32

server

connecting to

function example, 271–275

Perl script, 396

`VixHost_Connect` function and, 237–239

`Vmctl` object and, 169–171

disconnecting from, 239–240

registering virtual machine with, 163–164, 188–191, 243

server console

Connect to Host screen, 46, 47

console preferences, 62, 66–72

Home tab, 61–62, 355

host settings, 62–66

installation wizard, 40

Inventory pane, 58–59

launching manually, 52–53

Linux, 33

main pane, 59–61

screen, 57–58

Switch Host screen, 41

updating, 68

Windows, 23

`$server®get_last_error()` method, 413

Set Access Rights screen (New Virtual Machine Wizard), 102, 358

set, definition of, 160

setgid sticky bit, 95

Setup Manager Wizard (Sysprep)

answer file

saving, 371, 372

storing, 121

categories, 121

Completing Setup Manager screen, 371, 372

License Agreement screen, 120, 370

Name and Organization screen, 371

New or Existing Answer File screen, 119, 369

Type of Setup screen, 119, 370

- Setup Type screen, 20, 21**
 - shebang (#!), 153, 184**
 - shortcuts, ~directory, 24**
 - shutdown command, 115**
 - SID (security identifier), 116, 142**
 - simple file sharing, 400**
 - single-thread operations, 236–237**
 - 64-bit platform, 353**
 - snapshot**
 - as alternative to nonpersistent disk, 144
 - creating, 251–252
 - getting handle that represents, 254–255
 - platform testing with, 129, 130
 - removing, 255–256
 - reverting virtual machine to state present in, 256–258
 - root, determining number of, 253–254
 - use of, 134
 - snapshot errors, Programming API, 420–421**
 - snapshot handle**
 - description of, 222
 - properties, 224
 - Snapshots section (Options tab), 77–78**
 - software**
 - antivirus, 134
 - for virtual development machine, setting up for deployment, 367–374
 - software development, virtual machines for, 126–127**
 - software testing, virtual machines for, 128–130**
 - software virtualization, 2**
 - Solution Explorer (Visual Studio Team System),**
 - References tab, 332**
 - sound adapter, virtual, 75**
 - source control**
 - adding, 289–291
 - adding developmental project to, 285–288
 - configuring, 387
 - target environment, 353
 - Source Control Explorer (Visual Studio Team System), 322, 346, 348**
 - Specify Disk Capacity screen (New Virtual Machine Wizard), 107–108**
 - Specify Disk File screen (New Virtual Machine Wizard), 108–109, 364**
 - Specify the Settings for the Project Portal screen (New Team Project wizard), 317**
 - Specify the Source Control Settings screen (New Team Project wizard), 317**
 - Specify the Team Project Settings screen (New Team Project wizard), 316**
 - Split disk into 2 GB files option (New Virtual Machine Wizard), 363**
 - SSL certificate, 32, 37**
 - standard errors, Programming API, 418–420**
 - standards, enforcing, 133**
 - Start menu, 23**
 - starting. See also opening**
 - CCNet application, 292
 - virtual machine, 195–199
 - Startup/Shutdown options (New Virtual Machine Wizard), 102–103, 357–358**
 - Startup/Shutdown Options screen (New Virtual Machine Wizard), 102–103, 359**
 - Startup/Shutdown section (Options tab), 78–82**
 - state, definition of, 159**
 - state file location, configuring, 384–385**
 - storing**
 - answer file, 121
 - virtual disk, 108–109
 - string buffer, freeing, 235–236**
 - su – (switch user) command, 27**
 - subnet, placing development machines on different, 132**
 - Subnet window, 88**
 - Summary tab (Virtual Network Editor), 85**
 - support analysis, 131**
 - supported platform, 352**
 - suspending virtual machine, 249–250**
 - Switch Host option (Home tab), 62**
 - switch user (su –) command, 27**
 - switching view, 71–72**
 - synchronous method, 167**
 - synchronous operation, 226**
 - Sysprep (System Preparation Tool) utility**
 - blank fields and, 371
 - description of, 367
 - folder, creating, 368–369
 - freezing Windows guest with, 116–123
 - Reseal button, 374
 - Setup Manager, 119–121, 369–371
 - tool screen, 373–374
 - versions of, 368
 - System Properties screen, 210, 211**
 - System Startup and Shutdown Defaults screen, 83**
 - System Tools menu (Linux), 33**
- ## T
- tabs (main pane), 60**
 - tar (tape archive), 25**
 - target environment**
 - defining, 351
 - platform, 352
 - source and process control, 353
 - tools, 352
 - Task class, 333**
 - tasks to automate, 152**
 - team**
 - collaborating with on base image, 133
 - common environment for, 126–127
 - team build**
 - configuring, 322–328
 - testing, 328–330
 - Team Explorer pane (Visual Studio Team System), 314, 319**
 - Team Foundation Server, 319, 330**
 - team project**
 - adding development project to, 318–322
 - creating, 312–318
 - placing class library and batch file in, 342–347
 - teaming errors, Programming API, 421**
 - temporary object, creating, 236**
 - test bed, 8, 9**
 - testing. See also platform testing**
 - development image, 134–135
 - team build, 328–330
 - testing environment, isolating, 91**

testing image

- configuring as nonpersistent, 378–380
- description of, 138
- naming, 303–304
- recycling, 143–144
- setting up, 374–377
- test-winxppro-sp2 file, 302–304
- text console, installing VMware Tools and, 114**
- ThoughtWorks.CruiseControl.MSBuild.XMLLogger library, 392
- tilde (~)**
 - directory shortcut, 24
 - modifier, 35
- Title and Status area (main pane), 60**
- /tmp folder, 115
- tools, for target environment, 352**
- tools screen (Visual Studio Team System), 345**
- Tools* Connect to Team Foundation Server, 313**
- Tools* Folder Options, 400**
- ToolsLastActive property (VmCtl object), 166
- trigger, adding, 288–289**
- trigger block, configuring, 387–388**
- troubleshooting installation**
 - of guest operating system, 112–113
 - of VMware Server, 48–55
- Type of Setup screen (Setup Manager Wizard), 119, 370**

U

- undefined, in Perl, 191**
- uninstalling previous VMware product, 18**
- unregistering virtual machine, 244**
- updating**
 - after deployment, 133–134
 - console, 68
- upgrading**
 - kernel, 34
 - virtual hardware, 261–262
- USB port, virtual, 75**
- user context, running virtual machine under, 103**
- user rights to virtual machine**
 - categories of, 80
 - configuring, 93–96
 - levels, 92–93
- using **declarations, 333**
- utilities**
 - GZip (GNU zip) compression, 25, 140
 - Sysprep (System Preparation Tool), 116–123, 367–374
 - vmrun utility, 157
 - vmware-cmd utility, 156
- utilization of hardware, 8**

V

- VB.NET, scripting VMware Tools with, 283–284**
- .vbs file, 160
- VBScript, scripting VMware Tools with, 279–282**
- verifying**
 - connection to server, 187, 192
 - port 902 availability, 50
 - VmExecutionState, 178
- view, switching, 71–72**
- View tab (Folder Options dialog box), 401**
- View* Other Windows* Pending Changes, 321**
- View* Pending Checkins, 287**
- virtual BIOS, 74**
- virtual chip set, 74**
- virtual development machine**
 - deleting floppy drive, 365
 - renaming configuration file, 365–367
 - setting up, 355–364
 - setting up software for deployment, 367–374
- virtual disk**
 - creating, 360
 - pre-allocating, 13, 364
 - size of, 363
 - storing, 108–109
- virtual disk file, renaming, 108, 141, 363**
- virtual Ethernet card, 75**
- virtual floppy drive, 74**
- virtual graphics, 74**
- virtual hardware, upgrading, 261–262**
- virtual IDE drive, 74**
- virtual keyboard, 75**
- virtual machine**
 - connecting to, powering on, and copying file to, 271–275
 - creating, 98–109
 - deleting, 250–251
 - description of, 101
 - installing operating system onto, 109–113
 - iterating list of registered, and displaying state of, 171–173
 - naming, 138
 - opening, 245–246
 - for post-release maintenance, 131
 - powering off, 247–248
 - powering on, 111, 169–171, 246–247
 - registering
 - remotely, 270–271
 - with server, 163–164, 188–191
 - resetting, 248–249
 - for software development, 126–127
 - for software testing, 128–130
 - starting, 195–199
 - suspending, 249–250
 - unregistering, 244
 - user rights to, 80
- virtual machine handle**
 - description of, 222
 - properties, 223
- Virtual Machine Properties, Security tab, 94**
- virtual machine security, 92–93**
- Virtual Machine Settings window**
 - components of, 72–73
 - Floppy node, 365
 - Hardware tab, 109–110, 148
 - Options tab, 75–84, 141–142
- Virtual Machine Startup and Shutdown screen, 82**
- virtual memory, 74**
- virtual mouse, 75**
- Virtual Network Editor**
 - Automatic Bridging tab, 86
 - DHCP tab, 92
 - Host Virtual Adapters tab, 91

- Host Virtual Network Mapping tab, 87–90
- NAT tab, 92
- Summary tab, 85
- Virtual Network Settings window, opening, 84**
- virtual networking, 84–85, 360–361**
- virtual parallel (LPT) port, 75**
- virtual PCI slot, 74**
- virtual processor, 73**
- virtual SCSI device, 74**
- virtual serial (COM) port, 75**
- virtual sound adapter, 75**
- virtual USB port, 75**
- virtualization**
 - of development server, 9
 - host machine and guest, 6
 - hosted versus hypervisor architecture, 3–5
 - software, 2
- virtualization layer, 2**
- Visual C++, compiling against Vix with**
 - include statement and, 215
 - overview of, 209–210
 - pointing project
 - to header files, 216–218
 - to libraries used, 218–220
 - setting up project, 210–215
- Visual SourceSafe**
 - Add to SourceSafe dialog box, 286–287
 - installing, 285
 - logging into, 286
 - Pending Checkins page, 287
- Visual Studio IDE**
 - opening, 312
 - Pending Changes tab, 320, 321, 347
- Visual Studio Team System (VSTS). See also build integration, configuring**
 - adding new assembly to team build XML file, 347–349
 - build hooks, 330
 - coding build task, 333–337
 - continuous integration and, 278
 - copying binaries to virtual machine, 337–338
 - custom assembly, creating for integration, 330–333
 - description of, 151, 312
 - New Team Build Type Creation wizard, 323–327
 - New Team Project wizard, 315–318
 - placing class library and batch file in team project, 342–347
 - team build
 - configuring, 322–328
 - testing, 328–330
 - team project
 - adding development project to, 318–322
 - creating, 312–318, 319
 - VMIntegration class, 338–342
 - VMware Server integration, 330–349
- Vix. See Programming API (Vix)**
- Vix_CreateTemporaryObject() function, 236**
- Vix_Error object, 417–418**
- VIX_ERROR_CODE() function, 225**
- VIX_FAILED() function, 226**
- Vix_FreeBuffer() function, 235–236**
- Vix_GetErrorText() function, 234–235**
- Vix_GetHandleType() function, 233–234**
- Vix_GetProperties() function, 231–232**
- Vix_GetPropertyType() function, 232–233**
- VixHost_Connect() function, 237–239**
- VixHost_Disconnect() function, 239–240**
- VixHost_FindItems() function, 240–243**
- VixHost_RegisterVM() function, 243**
- VixHost_UnregisterVM() function, 244**
- VixJob_CheckCompletion() function, 227, 269**
- VixJob_GetError() function, 268–269**
- VixJob_Wait() function, 228–230, 268**
- VIX_OK error code, 225**
- VIX_PROPERTY_NONE property, 232**
- VIX_PROPERTY_VM_TOOLS_STATE property, 275**
- Vix_PumpEvents() function, 236–237**
- Vix_ReleaseHandle() function, 231**
- VIX_SUCCEEDED() function, 226**
- VixVM_CopyFileFromGuestToHost() function, 266–267**
- VixVM_CopyFileFromHostToGuest() function, 265–266**
- VixVM_CreateSnapshot() function, 251–253**
- VixVM_Delete() function, 250–251**
- VixVM_GetNumRootSnapshots() function, 253–254**
- VixVM_GetRootSnapshot() function, 254–255**
- VixVM_InstallTools() function, 258–259**
- VixVM_LoginInGuest() function, 262–263**
- VixVM_Open() function, 245–246**
- VixVM_PowerOff() function, 247–248**
- VixVM_PowerOn() function, 246–247**
- VixVM_RemoveSnapshot() function, 255–256**
- VixVM_Reset() function, 248–249**
- VixVM_RevertToSnapshot() function, 256–258**
- VixVM_RunProgramInGuest() function, 263–265**
- VixVM_Suspend() function, 249–250**
- VixVM_UpgradeVirtualHardware() function, 261–262**
- VixVM_WaitForToolsInGuest() function, 259–261**
- VmCollection object, 171–173**
- VmCOM API**
 - description of, 153–154, 155, 156
 - enumerations, 177–180
 - error codes, 409–411
 - future of, 207
 - objects, 161
 - VmCollection object, 171–173
 - VmConnectParams object, 161–162
 - VmCtl object, 164–171
 - VmQuestion object, 173–177
 - VmServerCtl object, 162–164
 - VmConnectParams **object, 161–162**
 - VmCtl object**
 - example, 168–171
 - methods, 166–168
 - .vmdk file, 101**
 - .vmdkREDO file, 101**
- VM_EXECUTION_STATE constant, 204**
- VmExecutionState enumeration, 177–178**
- VmExecutionState, verifying, 178**
- vmExecutionState_Stuck state, 173, 174–177**
- VM_EXECUTION_STATE_STUCK state, 200–204**
- \$vm\$get_last_error() method, 413–414**
- VMIntegration class, 337, 338–342**

VMnet0, 85, 86, 87

VMnet1, 85, 87, 89

VMnet8, 85, 89

VmPerl API

- constants, 204–207
- description of, 153, 154–156
- error codes, 413–415
- future of, 207
- modules, 185
- VMware::VmPerl::ConnectParams module, 185–187
- VMware::VmPerl::Question module, 199–204
- VMware::VmPerl::Server module, 187–191
- VMware::VmPerl::VM module, 191–199

VmPlatform **enumeration, 179–180**

VM_POWEROP_MODE **constant, 204–205**

VmPowerOpMode **enumeration, 178**

VM_PRODINFO_PLATFORM **constant, 206**

VM_PRODINFO_PRODUCT **constant, 206**

VmProdInfoType **enumeration, 179**

VmProduct **enumeration, 179**

VmQuestion **object, 173–177**

vmrun **utility, 157**

VmServerCtl **object, 162–164**

vmSetIP.cs **file, 375–376**

VM[®] Settings, 72

.vmsn **file, 101**

.vmss **file, 101**

vmTestBank.pl

code, 307–310

file, 404–407

VMware Infrastructure, 157

VMware Player, running base image with, 135

VMware Server

- ability to move virtual machines from one server to another, 7–8
- Administration guide, 12
- advantages of, 6–9
- development process and, 8–9
- increased utilization of hardware, 8
- multiple platforms available on single physical machine, 7
- parts of, 11
- rapid deployment and, 6–7

VMware Server Console

- Connect to Host screen, 46, 47
- console preferences, 62, 66–72
- Home tab, 61–62, 355
- host settings, 62–66
- installation wizard, 40
- Inventory pane, 58–59
- launching manually, 52–53
- Linux, 33
- main pane, 59–61
- screen, 57–58
- Switch Host screen, 41
- updating, 68
- Windows, 23

VMware Server Management Interface, installing

- on Linux, 34–38
- remote client via
 - on Linux, 43–48
 - on Windows, 38–42

VMware Software Development Kit (SDK), 157

VMware Tools

- authenticating and logging in using, 262–263
- batch files, 281
- description of, 69
- installing
 - on guest, 258–259
 - on Linux guests, 113–116
 - on Windows guests, 113
- polling guest machine to determine if running, 259–261

VMware Tools icon, 116, 376

VMware Tools Properties screen, Scripts tab, 281, 282, 376, 377

VMware Tools scripting

- with C# and VB.NET, 283–284
- overview of, 279
- with VBScript, 279–282

VMware VmPerl Scripting API, 32

vmware-cmd **utility, 156**

vmware-config.pl **script, 27**

vmware-install.pl **script, 25**

vmware.log **file, 101**

vmware-users **group, creating, 95–96**

VMware::VmPerl::ConnectParams **module, 185–187**

VMware::VmPerl::Question **module, 199–204**

VMware::VmPerl::Server **module, 187–191**

VMware::VmPerl::VM **module, 191–199**

.vmx **file, 93, 101**

.vmx.sav **file, 101**

VSTS (Visual Studio Team System). See also build integration, configuring

- adding new assembly to team build XML file, 347–349
- build hooks, 330
- coding build task, 333–337
- continuous integration and, 278
- copying binaries to virtual machine, 337–338
- custom assembly, creating for integration, 330–333
- description of, 151, 312
- New Team Build Type Creation wizard, 323–327
- New Team Project wizard, 315–318
- placing class library and batch file in team project, 342–347
- team build
 - configuring, 322–328
 - testing, 328–330
- team project
 - adding development project to, 318–322
 - creating, 312–318, 319
- VMIntegration class, 338–342
- VMware Server integration, 330–349

W**web management interface**

- description of, 37
- installing API via, 42
- installing remote client via
 - on Linux, 43–48
 - on Windows, 38–42
- Startup and Shutdown configuration, 80–82, 83

Web sites

- CruiseControl.NET, 285
- ThoughtWorks.CruiseControl.MSBuild
 - .XMLLogger library, 392
- VMware Server Administration guide, 12
- WinImage, 111
- Wrox, 278

Welcome screen

- New Team Build Type Creation wizard, 323
- New Virtual Machine Wizard, 356
- Visual SourceSafe, 286
- VMware Server, 19
- Win32 Application Wizard, 214

Windows

- configuring user access on, 94–95
- freezing guest with Sysprep, 116–123
- host operating system requirements, 15

remote client

- installing on, 38–42
- requirements for, 18
- VMware Server, installing on, 19–24
- VMware Tools, installing on, 113

Windows Firewall dialog box, Exceptions tab, 401–402**Windows 98 Second Edition, 374****Windows Scripting Host, 160****WinImage program, 111, 145****WinRAR format, 140****Win32 Application Wizard, 213, 214****working directory, 298****Workspace tab (console preferences), 68****workstation, imaging, 127****writing code from within development virtual machine, 9****Wrox Web site, 278****.wsf file, 160****X****X environment, restarting, 115****xinetd, installing, 50–51****XML file, 333, 347–348****XML logging, 298, 391–393**