Index

A
Abstract classes, 247–248
vs. concrete, 248–253
interfaces. See Interfaces
summary, 271–272
Abstraction in design, 156
Access modifiers
for delegates, 284–285
purpose of, 158
Activating COM objects, 484, 491–492, 530
Activator class, 517
ActiveX DLL projects, 485
Add New Item command, 422, 526
Add Reference dialog, 502–503, 505
AddHandler keyword, 314–318, 322
AddRef method, 496
AddressOf operator
for delegates, 287, 290, 293
for dynamic event binding, 316
Administrators group
for Assembly Cache Viewer, 452
for GACs, 448
ADO library, 37, 511
AL.EXE (Assembly Linker), 477–478
Aliases for namespaces, 53–54
Allocating memory, 13, 19–20, 102
Alternative array declaration syntax,
146–147
Ambiguities
with imported namespaces, 53–54
with modules, 170
Ampersands (&) for concatenation, 124, 133
Anakrino tool, 49
And operator
for flags, 378–379
operation of, 85–87
AndAlso operator, 86–87
Angle brackets (<>) for attributes, 11
app.config file, 422
AppBase directory
for configuration files, 419–421
for dependent assemblies, 16
for interop assemblies, 502
for private assemblies, 443–445
AppDomains, 17–19, 162
Application Configuration File option, 422
Application frameworks of FCL, 29–33
ApplicationException class, 348, 362–365
Applications
ASP.NET, 31–33, 420
component-based, 46
configuration files for, 419–422
console-based, 42–45
forms-based, 30–31, 61–62
machine.config for, 424–426
precompiling, 463
settings for, 422–424
Windows, 62–66
Applying attributes, 11
<appSettings> element, 421, 423, 425
AppSettings property, 423–424
Architecture of .NET Framework, 5–7
ArgumentException class, 350
ArgumentNullException class, 350
ArgumentOutOfRangeException class, 350, 360–361
ArithmeticException class, 350–351
Array class, 149–153
Array initialization lists, 138, 149
Arrays, 135–141
alternative declaration syntax for, 146–147
classes for, 147–153
defining element types in, 143–146
exceptions with, 348
multidimensional, 147–149
parameter, 189–190
ReDim Preserve for, 141–143
As clauses
  in method signatures, 187
  in strict type checking, 79–80
As New syntax, 71–72
ASP.NET
  applications in, 31–33, 420
  .NET Framework for, 35
  shared fields in, 162
Assemblies, 7–10, 413–414
  for configuring applications, 419–426
dependent, 9, 16
deploying. See Deployment
digital signatures for, 9, 431–442
d as DLL solution, 414–418
exceptions with, 349
from FCL, 24
format strings for, 430
IL in, 10
inheritance across, 213, 216–217
inspecting, 48–49
interop. See Interop assemblies
loading, 9, 13, 16, 443–446, 457–460
names of, 426–429
vs. namespaces, 54–55
relinking, 55, 59
strong names in, 431–433
summary, 480
versioning. See Versions
Assembly Binding Log Viewer utility, 460–461
Assembly Cache Viewer utility, 451–452
Assembly keyword, 11
Assembly-level attributes, 521–522
Assembly Linker (AL.EXE), 477–478
Assembly manager
  audit logs by, 461
  in CLR loading process, 458–460
  purpose of, 449
Assembly manifests, 9, 16
Assembly purgatory, 418
Assembly resolution process, 16
Assembly value in Registry, 528
<assemblyBinding> element, 445, 479
AssemblyDescription attribute, 521
<assemblyIdentity> element, 454, 472
AssemblyInfo.vb file, 427
AssemblyKeyFile attribute, 433, 521
AssemblyVersion attribute, 427, 432,
  467–469, 521
Assert method, 69
Assertions, 69
Assignment statements, 72–73
Asymmetric encryption, 435
Attacks
  on DLLs, 415
  safe mode for, 21
Attribute-based programming, 10
Attributes
  in COM, 521–527
  in .NET Framework, 10–11
Audit logs, 461
Authentication
  assembly manifest information for, 9
  managed execution for, 21
Authenticode, 438
Auto-initialized variables, 172
AutoDispatch class, 523–524
AutoDual class, 524, 526

B
Background compilation, 57
Backward compatibility
  of DLLs, 416–417
  in versioning assemblies, 464–467
Bang operator, 201
Banker’s rounding, 110
Base Class Library (BCL), 23
Base classes, 205–208
  chaining calls to, 229–231
  converting references to, 222
  inheritance in, 210–216
  in .NET Framework, 209–210
  versioning with, 212–213
Base types, 94
BCL (Base Class Library), 23
BeginInvoke method, 285–286
Binary compatibility mode, 467, 492
Binary option for string comparisons,
INDEX

130–131
Binary reuse, 46
BinarySearch method, 152
Binding
in COM, 484
early, 489–490
late, 497–499, 516–518, 532
delegates to instance methods, 289–291
dynamic
of events, 314–318, 322
operation of, 224–229
static
of events, 318–327
operation of, 224–226
<bindingRedirect> element, 471–476, 480
Bitwise copies, 390
Bitwise operations vs. logical comparisons, 85–87
Blocks in scoping rules, 74
Boolean values
with Option Strict, 84–85
in string comparisons, 131
in When clauses, 352
Bound, array, 135–136, 138
exceptions with, 348
with ReDim, 149
Boxing process, 114–118, 146
Break mode, 67
Breakpoints, 67
BSTR type, 510, 533
Buffer overflow attacks, 21
Buffers for strings, 126
Bugs
fixes for, 465, 468
strict type checking for, 79
Build numbers, 427, 467
asterisks for, 469
incrementing, 468
in version redirection, 472
ByRef keyword, 187, 384–387
Byte type, 104
in enumerations, 374
in narrowing conversions, 81–82
ByVal keyword, 187, 384–387

C
C++ language, function pointers in, 282–283
C# language, 7, 12
Caches
for <codeBase> elements, 456–457, 459
for deployment, 461–464
for performance, 35
Call chains, 98–99
from derived classes to base classes, 229–231
for exceptions, 358
Call stacks
for arrays, 136–137, 140, 144–145
in boxing process, 114–116
for copy-on-assignment, 383
exceptions with, 349
for method execution, 97–99
for parameter passing, 385–387
for strings, 123
for structure values, 388
for variables, 100–102
Callbacks
with delegates, 273–281, 291–294
for events, 274, 306
function pointers for, 282
multicasting for, 296–300
registering references for, 166
CAS (Code Access Security), 22, 456–457
Case sensitivity
in interoperability, 92
in string comparisons, 128, 130, 132
in XML, 421
Catch-all exception handlers, 351
Catch blocks, 76–77, 339–345
cleanup code in, 366
for exceptions
filtering, 350–352
inner, 357–359
system-defined, 360
nesting, 346
StackTrace property for, 355–356
When clauses in, 352–353
CBool operator, 83
CByte operator, 81, 83
CChar operator, 83, 106
CCWs (COM callable wrappers), 532–534
CDate operator, 83
CDbl operator, 83
in primitive type conversions, 109, 111
with unboxing, 117
CDec operator, 83, 111
CE devices, 36
Cell phones, 36
Certificates vs. signatures, 438–439
early, 489–490
late, 497–499, 516–518, 532
callable wrappers in
COM, 532–534
runtime, 509–512
DLLS for, 414–415
building and testing, 529–530
creating, 485–486
interface definition languages for, 487–489
registering, 527–528
Registry for, 490–491
type libraries for, 486–490
Visual Basic .NET, 518–519
error reporting in, 492–495
events in, 306
exceptions in, 512–514
interop assemblies for, 500–503
primary, 503–506
programming against, 506–509
for language independence, 90
limitations of, 90–91
memory leaks in, 398
objects in
activating, 484, 491–492, 530
lifetime of, 495–497
releasing, 514–516
OOP support in, 4
parameterized constructors in, 518–519
reference counting in, 394–395
scripting clients in, 497–499
summary, 534–535
type library exporter for, 519–520
version problems in, 524
COM+, 10
COM callable wrappers (CCWs), 532–534
COM tab, 502
Combine method, 294–297, 310
ComClass attribute, 525–527
COMException class, 512–513, 533
Common Language Infrastructure (CLI), 6–7, 23
Common Language Runtime. See CLR (Common Language Runtime)
Common Language Specification (CLS), 91–93
Common Type System. See CTS (Common Type System)
Compact Framework (CF), 5
Compacting managed heap, 396
Companies, public keys for, 428
Compare method, 128–129
CompareOrdinal method, 129, 131
Comparing
references, 102–103
strings
using equal signs, 130–132
using methods, 127–129
Compatibility
of DLLs, 416–417
with new features, 73
with polymorphism, 221–222
in versioning assemblies, 464–467
of Visual Basic .NET, 12
Compatibility Checker, 467
Compilers and compiling, 7, 41–43
background, 57
Const fields, 175
into IL, 10
JIT, 15–16, 21, 47, 461
managed, 8, 13
name ambiguities in, 53–54
optional parameters, 188, 194
type definitions in, 95–96
Component Services namespace, 111, 130
Component-based applications, 46
Component libraries
for exception classes, 364
loading, 16
writing and using, 45–47
Component metadata
in assemblies, 8–9
CLR inspection of, 15
for optional parameters, 188
Component Object Model. See COM (Component Object Model)
Concat method, 96, 124
Concatenating strings, 124–127
Concrete classes, 248–253
Conditions in When clauses, 352–353
.config files, 419
<configSections> element, 425
<configuration> element, 421, 423
Configuration files for applications, 419–422
ConfigurationException class, 424
ConfigurationSettings class, 423
Conflicts, name, 49–52, 75, 415
Console applications, 29, 56
collection files for, 422
developing, 42–45
Console class, 29
Const fields
  in contracts, 465
  working with, 174–177
Constants
  for Char type, 107
  in Microsoft.VisualBasic assembly, 25, 27–29
Constructors, 177
  in assemblies, 48
  in COM, 518–519
default, 178–181, 214–215, 381
for delegates, 285
exceptions with, 360–361
inheritance in, 213–216
initializers in, 156, 183–185
instance, 177–183
in interfaces, 254
overloading, 180–183
shared, 185–186
for structures, 381
for user-defined exceptions, 363
Containing assemblies, 216
Containment, 209
Contracts
  with DLLs, 416
  for interfaces vs. classes, 259–261
  in versioning, 465–467
Control class, 219–223
Controls, creating, 65–66
Conversions
  with Boolean type, 84–85
  in COM, 510–511, 533
  with delegates, 295
  for enumerations, 375
  exceptions with, 348
  with interface members, 260
  with Object class, 112–113
  in polymorphism, 221–224
  with primitive types, 108–112
  with strict type checking, 81–84
  with strings, 110–111, 132–133
  with unboxing, 116
Copies
  deep, 151, 392–393
  shallow, 151, 389–392
Copy method for arrays, 151
Copy-on-assignment, 383, 388
Core CTS types, 104
  Object class, 112–121
  primitive, 104–112
  strings. See Strings and String class
Country settings, 427–428
Creatable classes, 248
Creatable wrapper types, 508
CreateInstance method, 517–518
Cross-assembly inheritance, 213, 216
Cross-platform capabilities, 5–6
  for JIT compiling, 21
  library support in, 23
Cryptography, 434–437
CShort operator, 83, 117
CSng operator, 83
CStr operator, 83, 133
.ctor keyword, 48, 180
CTS (Common Type System), 89–90
  arrays. See Arrays
  CLS for, 91–93
  concepts in, 93–94
  inheritance in, 94–96
  motivation for, 90
  Object class in, 112–121
  primitive types in, 104–112
  reference comparisons in, 102–103
  shared members vs. instance members in, 96–97
  strings. See Strings and String class
  summary, 153
  threads, method execution, and calls stacks in, 97–99
  value types and reference types in, 99–102
CType operator, 83
  exceptions with, 348
  with interfaces, 260
  with polymorphism, 223
  in primitive type conversions, 109
  in string conversions, 133
  with unboxing, 117–118
Culture settings and information
  for assemblies, 9, 427–428
  in format strings, 430
  in string comparisons, 128–129, 132
CultureInfo class, 128–129
Curly braces ({})
  for array elements, 138
  in format strings, 134
Currency type
  in formatting code, 134–135
  support for, 106
  Custom attributes, 11
Custom exception classes, 362–365
Customized event parameters, 333–337

D
Data assembly, 26, 37, 58
Dataset namespace, 52
Date keyword, 107
Dates in formatting code, 134
DateTime type, 107–108
DCOM (Distributed COM) protocol, 35
DDB function, 28
Debug.Assert method, 69
Debug class, 68–70
Debug configurations, 66–67
Debug mode, 355
Debug.WriteLine method, 68–69
Debugging
AppDomains, 17
.NET Framework for, 34
projects, 66–70
strict type checking for, 79
Decimal type, 106, 109
Declaring
arrays, alternative syntax for, 146–147
methods, 172, 195–196
namespaces, 52
variables, 75, 79, 100–101
Decoupling classes, 259
Deep copies
for arrays, 151
for objects, 392–393
Default keyword, 200–201
Defaults
AppDomains, 18
COM interfaces, 486, 524
constructors, 178–181
inheritance in, 214–215
for structures, 381
method implementations, 226
namespaces, 50
optional parameter values, 188, 194
properties, 65, 199–202
versioning policies, 470
Defining
array element types, 143–146
events, 308–311
interfaces, 254–255
multiple variables, 71
Defragmenting memory, 396
Delaying digital signatures, 439–442
Delegate keyword, 284
Delegates, 273, 284–286
binding to instance methods, 289–291
for callback implementation, 291–294
callback notifications for, 273–281
creating, 286–289
for event handlers, 308–311
events for, 306–307
GetInvocationList for, 300–302
memory for, 331
 multicasting for, 294–300, 306–307
summary, 303
Demeanor obfuscator, 49
Dependencies, COM
late binding for, 517
references to, 504
Dependent assemblies, 9, 16
exceptions with, 349
references for, 59
version numbers for, 470
<dependentAssembly> element, 454, 472, 479
Deployment
assemblies, 442
Assembly Binding Log Viewer for, 460–461
CLR loading process for, 457–460
with <codeBase> elements, 453–457
in Global Assembly Cache, 447–453
as migration challenge, 37
native image caches for, 461–464
.NET Framework for, 34
private, 443–447
publisher policy files, 478
Derived classes, 206–208
chaining calls from, 229–231
converting references to, 222
replacing methods in. See Replacing methods
Derived types, 94
Design
of classes, 155–156
.NET Framework for, 34
with Object class, 119–121
with object finalization, 401–402
with overrideable methods, 230–232
with shared members, 163–166
Desktop applications, precompiling, 463
Destructors, 496
INDEX

Developers, public keys for, 428
Diagnostic logs, 350
Diagnostics namespace, 68
Digital signatures
  with assemblies, 9, 431–442
delaying, 439–442
  vs. digital certificates, 438–439
managed execution for, 21
Dimensions, array, 147–149
DirectCast operator, 116–118, 260
Directories, dependent assemblies in, 16
DirectoryServices assembly, 26
Disabling
  verification checks, 441
  version redirection, 479–480
DispIDs, 498–499
Disposable objects, 403–406
Dispose method, 404–410
disposing parameter, 410
Distributed applications, 35
Distributed COM (DCOM) protocol, 35
Distribution. See Deployment
DivideByZeroException class, 350–351
DLL hell, 418
DLLs. See also Libraries
  for COM. See COM (Component Object Model)
    downloading from Internet, 22
    for exception classes, 364
    problems with, 414–418
dollar signs ($) in type conversions, 111
DOS operating system, 2
Double type, 104, 106
  converting Decimal to, 109
  in narrowing conversions, 82–83
Down-casting in polymorphism, 222
Download caches, 456–457, 459
Downloading libraries, 16, 22
Duplicating array objects, 151
Dynamic binding
  of events, 314–318, 322
  operation of, 224–229
Empty strings, 123–124
Empty value, 124
Encapsulation
  assemblies for, 9
  with inheritance, 212–213
  with types and members, 157–159
Encryption, 434–437
End Try statement, 342–343
EndInvoke method, 285–286
EnterpriseServices assembly, 26
Enum class, 375–376
Enum construct, 372
Enumerations, 372
  base types for, 210
  creating, 373–377
  for error codes, 513
  as flags, 377–379
Environment variables, 17
Equal signs (=)
  in assignment statements, 72
  for strings, 130–132
Equals method
  in COM, 524
  for strings, 127–129
Err.Description message, 495
Err.Number error code, 495
Err.Raise syntax, 494
ErrorCode property, 513
Errors. See also Exceptions; Structured exception handling
  COM reporting of, 492–495
  Visual Studio .NET messages for, 57
Event-driven model, 62
Event handlers, 306–307
  with callbacks, 274
  creating and registering, 314–328
dynamic binding in, 314–318, 322
  static binding in, 318–327
Event keyword, 308
Event sources, 306–307
EventHandler delegate, 328–333
EventLog class, 344
Events, 305–306
  with callbacks, 274, 306
  customized parameters for, 333–337
defining, 308–311
  history of, 305–307
  in inherited classes, 327–328
  raising, 312–314
  summary, 337
System.EventHandler delegate for, 328–333
Exception class, 347–349
in Catch blocks, 351–352
members of, 354–357
Exceptions
with CCWs, 533
classes for, 346–350
in COM, 493–494, 512–514
handling. See Structured exception handling
with object finalization, 402
Executable instructions
in assemblies, 8
IL for, 10
Execution
managed, 19–22
method, threads and call stacks in, 97–99
monitoring, CLR for, 13
ExecutionEngineException class, 349
Explicit references to external COM dependencies, 504
Extending base class implementations, 230
Extensions, file, 8
External COM dependencies, explicit references to, 504

F
Factory methods, 166–167
FCL (Framework Class Library), 1–2, 22
Application frameworks of, 29–33
benefits of, 34
Microsoft.VisualBasic assembly for, 25–29
system-supplied class libraries in, 24–26
Fields, 172–174
in classes, 156
Const and ReadOnly, 174–177
for events, 307
inheriting, 210–213
initializing, 172–173, 179–186
in interfaces, 254
overriding, 227
in structures, 380
File extensions, 8
File headers, 13
File name conflicts, 415
File Transfer Protocol (FTP), 34
FileNotFoundException class, 459–460
Filtering exceptions, 350–352
Finalization, objects, 399–402
Finalize method, 230–231, 399–403
for COM objects, 514
vs. Dispose, 405
with Dispose, 406–410
Finally blocks, 365–370
Flags, enumerations as, 377–379
Flags attribute, 377–379
Floating-point types, 104
For loops for arrays, 150–151
For Each loops
for arrays, 150–151
for delegates, 301
for strings, 122
Form class, mouse events in, 334
Form designers, 30
Format method, 134–135
FormatException class, 112
Formatting codes, 134
Formatting strings, 134–135, 430
Forms applications, 30–31
creating, 61–62
event handlers in, 329–330
Framework Class Library (FCL)
Application frameworks of, 29–33
benefits of, 34
Microsoft.VisualBasic assembly for, 25–29
system-supplied class libraries in, 24–26
FreeBSD, 7
Friend access modifier, 157–158
for constructors, 183
for delegates, 284
for inheritance, 211, 217
in interfaces, 255
for nested classes, 203
with public members, 159
Friendly names
for assemblies, 9, 426
in compilation, 47
in format strings, 430
FromObject method, 116
FromString method, 111
FTP (File Transfer Protocol), 34
Fully qualified names, 50
Function keyword, 187
for delegates, 284
for event handlers, 310–311
Function pointers in C++, 282–283
FUSION.DLL component, 449–451
FUSLOGVW.EXE utility, 460–461
INDEX

G
GAC (Global Assembly Cache), 442
assembly management within, 449–453
benefits and drawbacks of, 447–449
deploying assemblies in, 447–453
for publisher policy files, 478
GACUTIL.EXE utility, 450–451, 505
Garbage collection, 393–396
CLR for, 13
for object finalization, 399–402
operation of, 396–397
programmatic control of, 402–403
vs. reference counting, 397–399
GC class, 402–403
Get accessors, 196–197
Get blocks, 196–199
Get methods with static event binding, 324
GetBaseException method, 359
GetHashCode method, 524
GetIDsOfNames method, 498, 518
GetInvocationList method, 300–302
GetType method, 376, 524
GetTypeFromProgID method, 517
Global Assembly Cache (GAC), 442
assembly management within, 449–453
benefits and drawbacks of, 447–449
deploying assemblies in, 447–453
for publisher policy files, 478
Global namespaces, 50
Globally unique IDs. See GUIDs (globally unique IDs) in COM
Graphical user interfaces (GUIs), 3
Guarded blocks, 342
GUI-based applications, precompiling, 463
GUID attribute, 506–507
GUIDGEN.EXE utility, 487, 521
GUIDs (globally unique IDs) in COM, 486–487
in ComClass attribute, 525–526
generating, 521
late binding for, 517
for type library exporter, 520–521
GUIs (graphical user interfaces), 3

H
&H prefix, 104
Hacker attacks on DLLs, 415
Handlers, 306–307
with callbacks, 274
creating and registering, 314–328
dynamic binding in, 314–318, 322
static binding in, 318–327
Handles clause, 319, 327, 332
“Has a” relationships, 209
Hash functions, 434
Hash values, 434–436
Headers for managed compilers, 13
Heap. See Managed heap
Help feature, 55
Helper methods, 25
Hexadecimal numbers, 104
Hide-by-name, 241
Hide-by-signature, 243
Hiding members, 242, 261–264
Hierarchy, inheritance, 94–95, 208
in Forms framework, 219
for user-defined exceptions, 364–365
href attribute, 454–455
HRESULT type, 492–495, 512–514, 533
HRESULT type, 492–495, 512–514, 533
HTML (Hypertext Markup Language), 32
Hypertext Markup Language (HTML), 32

I
-i switch in GACUTIL, 450
ICloneable interface, 253, 388–389
IComparable interface, 152
IDE (integrated development environment)
benefits of, 39
as migration challenge, 37
Idiosyncrasies in Visual Basic, 73–74
IDispatch interface, 498–499, 518, 524
IDisposable interface, 253, 404
IDLs (interface definition languages), 487–489
IDs, 487, 490, 492
and ComClass attribute, 526
late binding for, 517
for type library exporter, 520
IIS (Internet Information Services), 31
IIS6, .NET Framework for, 35
IL (intermediate language), 10
in assemblies, 49
boxing process in, 115
JIT compiler for, 461
ILDASM.EXE utility, 48–49
Image caches, 461–464
Immutable data, 125
Implementing
callbacks, 291–294
interfaces, 255–257, 262–268
methods, 186
Implements keyword, 255–257, 262–268
Implicit conversions to Object class, 112–113
Importing namespaces, 52–56, 60–62
Imports statement, 52–55, 60–62
Inconsistencies
in Visual Basic, 73–74
with wrapper methods, 28
Index numbers with arrays, 135–136, 348
Indexers, 200
IndexOf method, 96, 152, 353
IndexOutOfRangeException class, 348
Inheritance, 205
from classes, 205–209
limiting to containing assemblies, 216–217
members, 210–216
.NET Framework base classes in, 209–210
in CTS, 94–96
in interfaces, 253, 264–267, 269–271
and polymorphism, 218–224
replacing methods in. See Replacing methods
in structured exception handling, 347
in structures, 380, 388
summary, 245
in versioning, 466
WithEvents fields in, 327–328
Inheritance hierarchy, 94–95, 208
in Forms framework, 219
for user-defined exceptions, 364–365
Inherits keyword, 95, 207, 264–265
INI files, 419
Initializing
arrays elements, 138–139, 149
CLR sessions, 14–15
with constructors, 156, 183–185
fields, 172–173, 179–186
Const, 174
ReadOnly, 176–177
structure, 381
members, 156
primitive types, 104
variables, 71
Inner exceptions, 357–359
InnerException property, 359
InProcServer32 key, 490, 528
Input parameters, validating, 361
Inspecting assemblies, 48–49
Installing assemblies, 450
Instance members
binding delegates to, 289–291
combining shared members with, 166–168
constructors, 177–183
declaring, 172
memory for, 161
properties, 198
vs. shared members, 159–172
InStr function, 27–28
int keyword, 51
Integer type, 51, 104
Integers, 104
in enumerations, 374
in interoperability, 92
in narrowing conversions, 81–83
Integrated development environment (IDE)
benefits of, 39
as migration challenge, 37
Integration of applications, 35
Integrity checks, 361–362
Intellectual property, protecting, 49
IntelliSense feature
for COM, 490–491
for enumerations, 375
for Me, 165
MSVB7.DLL for, 57
for references, 58
for StandardModule, 169
Interface construct, 254
Interface definition languages (IDLs), 487–489
InterfacesDual value, 525
Interfaces, 253
COM, 484, 486
contracts for, 259–261
defining, 254–255
implementing, 255–257
with inheritance, 253, 264–267, 269–271
in loosely coupled designs, 279
members of
hiding and renaming, 261–264
mapping, 255–256
names of, 267–268
multiple, 267–269
programming against, 258–259
structures for, 380
summary, 271–272
visualizing, 261
InterfaceType attribute, 523, 525–526
Intermediate language (IL), 10
  in assemblies, 49
  boxing process in, 115
  JIT compiler for, 461
Internet, downloading libraries from, 16, 22
Internet Information Services (IIS), 31
Interop assemblies, 500–503
  attributes for, 521–522
  building and testing DLLs for, 529–530
  primary, 503–506
  programming against, 506–509
Interoperability, 36
  CLS for, 92
COM. See COM (Component Object Model)
in managed languages, 13
InteropServices namespace, 512
Interpreters, script, 497–499
InvalidCastException class, 111, 348
InvalidOperationException class, 349, 358–359, 362
Invoke method
  for event handlers, 310–312
  for late binding, 518
  for scripting interpreters, 499
IRR function, 28
Is-a relationships
  for inheritance, 208–209, 222
  with interfaces, 260
Is operator
  with interfaces, 261
  for reference comparisons, 102–103
IScreenPainter interface, 267–268
IsDefined method, 376–377
IsMissing function, 188
Isolation in AppDomains, 17
ItemChangedEventArgs class, 334
ItemChangedEventHandler delegate, 334–335
IUnknown interface, 495–496

K
Keyboard shortcuts for debugging, 67–68
/keyf switch in AL.EXE, 478
/keyfile switch in TLBIMP.EXE, 501
KeyPressEventArgs class, 334
KeyPressEventHandler class, 334
Keys
  for assemblies, 9, 428–429
  for encryption, 434–437
  for format strings, 430
  for signatures, 439–441
  for strong names, 431–432

L
Languages, assembly settings for, 427–428
LastIndexOf method, 152
Late binding
  in COM, 497–499, 516–518, 532
  problems with, 80–81
Lazy initialization, 71–72
Length
  of arrays, 135, 142
  of strings, 124–125
Length property
  for arrays, 149–150
  for delegates, 301
  for strings, 124
LibIDs, 487, 490, 520
Libraries
  for attributes, 11
  for COM, 486–490
  in cross-platform development, 23
DLL. See DLLs
downloading, 16, 22
  for exception classes, 364
FCL. See FCL (Framework Class Library)
loading, 16
  publisher policy files for, 476
  writing and using, 45–47
Lifetime of objects, 393–396
  in COM, 495–497, 514–516, 534
disposable objects in, 403–406
Dispose and Finalize in combination, 406–410
finalization in, 399–402
  reference counting in, 394–396
  shared fields, 162
Lightweight devices, 36
/link switch in AL.EXE, 477
Linux operating system, 7
List box controls, 65–66
Listeners
  for callbacks, 276, 279
  function pointers for, 282
Load events, delegates for, 329–331
Loading
  assemblies, 9, 13, 457–460
    component DLLs, 16
    private, 443–446
CLR, 13–14
Local variables, memory for, 100–102
Location independence of assemblies, 442
Log Failures option, 460
Logical comparisons, 85–87
Logs
  audit, 461
    for diagnostic information, 350
    in exception handling, 344
Long type, 104, 374
Loops
  for arrays, 150–151
  for delegates, 301
  for strings, 122
Loosely coupled designs, 279
Low-level instructions, 10
Low-level memory-management, 19–20
Lower array bounds, 135–136

M
machine.config file
  <codeBase> element for, 455
    contents of, 424–426
  for publisher policy files, 476
  for version redirection, 474–475, 480
Machine-level assembly code, 10
Main method, 43, 45
Major version numbers, 467
  for compatibility, 468
    in version redirection, 472–473
Managed code, 5, 7–10, 18–19
Managed compilers, 8, 13
Managed DLLs, registering, 527–528
Managed execution, 19–22
Managed heap
  for arrays, 136–137, 140, 144–145
    with boxing process, 114–116
    compacting, 396
  for copy-on-assignment, 383
garbage collection for. See Garbage
  collection
    for objects, 102, 113–114
    for parameter passing, 385–387
    for strings, 121, 123
    for structure values, 388
Managed languages, choosing, 12–13
Managed objects, activating, 530
Managed threads, 97
Managed types. See CTS (Common Type
System)
Managed wrapper interfaces, 508
Manifests, 9, 16
Mapping interface members, 255–256,
  262–263
Marshal class, 515
Marshaling, 510–511
Math class, 163
MCCORWKS.DLL, 14
Me keyword
  vs. MyBase and MyClass, 235
  with shared methods, 164–166
Me.New calls, 182
Members, 172
  abstract, 251
  accessibility to, 157–158
    in assemblies, 49
    in classes, 156
      constructors, 177–186
      fields, 172–177
      methods, 186–195
      nested types, 202–203
      properties, 195–202
      shared vs. instance, 159–172
      encapsulating, 157–159
      hiding, 242, 261–264
      inheriting, 210–216
      initializing, 156
      instance. See Instance members
        mapping, 255–256
Memberwise copies, 390
MemberwiseClone method, 390
Memory, 371
  allocating, 13, 19–20, 102
    for arrays, 136–137, 140, 144–145
    for COM objects, 495–497, 514–516
    compacting, 396
    for delegates, 331
    exceptions with, 349
    function pointers for, 282–283
garbage collection for. See Garbage collection
for instance fields, 161
for local variables, 100–102
for objects, 102, 113–114
for parameter passing, 385–387
for precompiling assemblies, 463
for shared fields, 161, 167
for strings, 121–123, 126
for structure values, 388
Memory buffers, 126
Memory leaks, 398
Message property, 353–354
MessageBox class, 64
Metadata
in assemblies, 8–9, 49
CLR inspection of, 15
for optional parameters, 188
Metadata-only members, 197
Method signatures, 186–187
in overloading, 190–192
in overriding, 228
Methods, 172, 186–187
abstract, 251
binding delegates to, 289–291
in classes, 156
compiling, 15–16
inheriting, 210–213
in interfaces, 254, 257
in Microsoft.VisualBasic assembly, 25, 27
optional parameters for, 187–189
overloading, 190–194
vs. optional parameters, 194–195
with polymorphism, 220
overriding, 226–229
parameter arrays for, 189–190
replacing. See Replacing methods
in structures, 380
threads and call stacks for, 97–99
Microsoft .NET Framework Configuration Administrative Tool, 447, 455–456, 473
Microsoft .NET Framework Redistributable, 5
Microsoft Installer technology, 452
Microsoft Transaction Server (MTS), 10
Microsoft.VisualBasic assembly, 25–29, 58
Microsoft.VisualBasic.dll, 47
Microsoft.VisualBasic namespace, 525
Microsoft Windows Server 2003, 6
Migration to .NET Framework challenges in, 36–38
motives for, 33–36
of unmanaged code, 483
Minor version numbers, 427, 467
for compatibility, 468
in version redirection, 473
Minus signs (-) in assignment statements, 72
Mobile code
in download caches, 456
permissions for, 22
Module type, 168–171
Monitoring execution, 13
Mono project, 7
Monolithic applications, 46
Mouse events, 334
MouseEventHandler delegate, 334
MSCORCFG.MSC tool, 447
MscorEE.dll file, 13–14, 528
mscorlib assembly, 24, 26, 47, 58
MSCORSVR.DLL file, 14
MsgBox method, 64
MSI files, 452–453
MSVB7.DLL, 57
MTS (Microsoft Transaction Server), 10
MulticastDelegate class, 285, 294–296
Multicasting
for callbacks, 296–300
for delegates, 294–296, 306–307
Multidimensional arrays, 147–149
Multilistener callback notifications, 280–281
Multiple inheritance, 209
Multiple interfaces, 267–269
Multiple languages, working with, 57
Multiple variables, 71
Multiple versions. See Versions
Multithreading, 273
MultiUse class
for COM DLLs, 485, 490, 496
late binding for, 517
MustInherit keyword, 250–252
MustOverride keyword, 251–252, 254
MyBase keyword
for events, 327
for member access, 229
vs. MyClass and Me, 235
MyBase.New syntax, 215–216
MyClass vs. MyBase and Me, 235
N
Named values, enumerations for, 373–377
Names
for assemblies, 426–429, 503–504
in assembly manifests, 9, 16
for attributes, 11
for command buttons, 64
of DLLs, 415
for fields, 172
in format strings, 430
for get and set methods, 197
for handler methods, 329
for interface members, 261–264, 267–268
in method signatures, 186
and scope, 75
strong. See Strong names
Namespace keyword, 50
/namespace switch in TLBIMP.EXE, 501
Namespaces, 49–52
vs. assemblies, 54–55
declaring, 52
importing, 52–56, 60–62
root, 59–60
in Visual Studio .NET, 58–61
Narrowing conversions
in primitive type conversions, 108–109
with strict type checking, 81–83
Native image caches, 461–464
Native keywords, 51
Nesting
namespaces, 51–52
Try statements, 346
types, 202–203
.NET Framework, 1–2
AppDomains in, 17–19
architecture of, 5–7
attributes in, 10–11
CLR environment in, 13–16
FCL in. See FCL (Framework Class Library)
history of, 2–5
managed code and assemblies in, 7–10
managed execution in, 19–22
managed languages for, 12–13
migration to
challenges in, 36–38
motives for, 33–36
summary, 38
.NET Framework Configuration
Administrative Tool, 447, 455–456, 473
.NET Remoting, 35
Networked environments, 35
Neutral culture setting, 428, 430
New operator
for constructors, 178–179
for delegates, 287
for enumerations, 374–375
for strings, 121
for structures, 381–382
with wrappers, 509
newVersion attribute, 472, 475
NGEN.EXE utility, 462
Nimda virus, 21
No-arg constructors, 178
Non-overridable methods, 251
Nondeterministic finalization, 401
Nonparameterized constructors, 178
Nonrecoverable exceptions, 349–350
Not operator, 85
NotAFiniteNumberException class, 350–351
Nothing value
for AppSettings, 423
for array elements, 139
in COM, 514–515, 534
for delegates, 295
exceptions with, 348, 359
for Objects, 114
in reference counting, 395
for references, 102
for strings, 123
in structure fields, 381
Notifications
callback, 273–281
function pointers for, 282
NotInheritable keyword, 207, 217, 234
NotOverridable keyword, 232–234
NullReferenceException type, 124, 348
Numeric values
converting to strings, 110–111, 133
in enumerations, 374
O
Obfuscators, 49
Object Browser, 55
Object class, 112
for arrays, 145–146
boxing process for, 114–118
conversions with, 112–113
default constructors in, 214, 216
designing with, 119–121
in inheritance hierarchy, 94–95, 209–210
as reference type, 113–114
with strict type checking, 80, 112–113
as variant type, 119
Object graphs, 173–174
Object-oriented programming (OOP), 3
in COM, 4
as migration challenge, 37
support for, 34
Object references for delegates, 290
ObjectDisposedException class, 405–406
Objects, 371
assigning to reference variables, 73
cleanup code for, 404
cloning, 388–393
garbage collection for. See Garbage
collection
lifetime of. See Lifetime of objects
managed heap for, 102, 113–114
from reference types, 99
scope of, 17
in structures vs. classes, 387
summary, 411
oldVersion attribute, 472, 475–476
OLEVIEW.EXE utility, 488–489
One-dimensional arrays, 148
One-way hash functions, 434
OOP (object-oriented programming), 3
in COM, 4
as migration challenge, 37
support for, 34
Open standards for Web services, 36
Operating system independence, 10
Operators, new, 72
Optimization
AppDomains for, 18
of Const fields, 175
precompiling for, 461–463
Option Compare option, 130–132
Option Explicit option, 78–79
Option Strict option, 78–79
Boolean values under, 84–85
with late binding, 518
Optional keyword, 194, 466
Optional parameters
in contracts, 465–466
vs. method overloading, 194–195
for methods, 187–189
Or operator
for flags, 378
operation of, 85–86
OrElse operator, 86
/out switch
in AL.EXE, 477
in TLBIMP.EXE, 501
in VBC.EXE, 44
OutOfMemoryException class, 349
Overflow errors in narrowing conversions, 81–82
OverflowException class, 341–345, 350–351
Overloading
constructors, 180–183
methods, 156, 190–193
vs. optional parameters, 194–195
with polymorphism, 220
properties, 201
shadowing in, 240–245
Overloads keyword, 243–244, 410
Overridable keyword, 228, 266
Overridable methods
with abstract classes, 250–251
design issues with, 230–232
shadowing, 240
Overrides keyword, 228
Overriding
Dispose, 410
methods, 226–229
P
Parameter arrays, 189–190
Parameter lists
in method signatures, 186
in overloading functions, 190–192
Parameterized constructors, 178–179, 181
in COM, 518–519
for structures, 381
Parameterized properties, 199–202
Parameters
in contracts, 465
for delegates, 286–288
event, 333–337
for EventHandler delegates, 328
with Object class, 120
optional
in contracts, 465–466
vs. method overloading, 194–195
for methods, 187–189
passing
to methods, 187
in structures, 384–387
as Visual Basic change, 74
on stack frames, 98
for user-defined exceptions, 363
validating, 361
Parent classes, 206
Parentheses ()
for arrays, 146–148
for parameters, 74
Parse method
in primitive type conversions, 111–112
as shared, 105–106
Passing parameters
to methods, 187
in structures, 384–387
as Visual Basic change, 74
Passing references, 166
Pasting GUIDs, 465, 468
.pdb files, 66–67
Performance
caching and state management for, 35
with Finalize, 406–407
with garbage collection vs. reference
counting, 397
with late binding, 499
with lazy initialization, 71–72
with static binding, 226
with string building, 126
with wrapper methods, 28
Periods (.) for namespaces, 52
Permissions, limitations of, 22
PIAs (primary interop assemblies), 503–506
Placeholders in format strings, 134–135
Platform boundaries, applications across, 35
Platform independent code, 10
Pluggable handler methods, 331
Plus signs (+)
in assignment statements, 72
for concatenation, 133
Pocket PCs, 36
Pointers in C++, 19–20, 282–283
Policies, versioning, 469–471
Policy files, publisher, 474–476
building, 476–478
deploying, 478
disabling version redirection from, 479–480
Polymorphism, 218–221
with abstract classes, 250
conversions in, 221–224
decoupling of classes for, 259
delegates for, 331
in loosely coupled designs, 279
with structures, 388
Popping stack information, 97
Porting
existing code, 12
Visual Basic 6 projects, 483–484
wrapper methods for, 27
Pound signs (#) for dates, 107
Power Users group
for Assembly Cache Viewer, 452
for GACs, 448
PPmt function, 28
Precision in conversions
narrowing, 82–83
primitive type, 109
widening, 83
Precompiling assemblies, 461–463
PrefJITing, 462
Preserve keyword, 141–143
Previous delegates, 295
Primary interop assemblies (PIAs), 503–506
/primary switch in TLBIMP.EXE, 504
PrimaryInteropAssembly attribute, 504
Primitive types, 104–108
converting between, 108–112
vs. strings, 122–123
Private access modifier, 157–158
in assemblies, 49
for Const fields, 176
for constructors, 183
for inheritance, 210, 212
for nested classes, 203
Private assemblies, 16
deploying, 443–447
loading, 443–446
Private keys
for encryption, 435–437
for publisher policy files, 477
for signatures, 439–441
for strong names, 431–432
PrivatePath attribute, 445
<probing> element, 444–445
Probing process, 443–444
Processes
vs. AppDomains, 17
managed code in, 18–19
Processor independence, 10
INDEX

ProgID attribute, 522
ProgIDs, 490
Programmable entities, 17
Programming contracts
interfaces as, 253
in versioning, 465–467
Programming models, COM vs. CLR, 486
Project menu, 58
Project Properties dialog, 56
for COM, 530
for root namespace, 59–60
for strict type checking, 78
for string comparisons, 130
Project References dialog, 531
Projects, 56
debbuging, 66–70
references to, 58–59, 531
Promotion rules, 192
Properties, 65, 172, 195–199
abstract, 251
inheriting, 210–213
in interfaces, 254, 257
names for, 262–263
overloading, 201
overriding, 227–228
parameterized and default, 199–202
shadowing, 240–245
in structures, 380
Property sheets, 61
Protected classes, 157–158, 203, 210–212
Protected Friend classes, 157, 211
Protection, safe mode for, 21
Public access modifier and fields, 45, 47, 157
for constructors, 214, 285
in contracts, 465
for delegates, 284–285
with Friend types, 159
for inheritance, 211
in interfaces, 254–255
limitations of, 195
for nested classes, 203
Public key tokens, 429
with GAC, 451
inspection for, 460
Public keys
for assemblies, 9, 428–429
for encryption, 434–435
in format strings, 430
for publisher policy files, 477
for signatures, 439–440
for strong names, 431–432
Publisher policy files, 474–476
building, 476–478
deploying, 478
disabling version redirection from, 479–480
<publisherPolicy> element, 479–480
Pushing stack information, 97

R
-R switch in SN.EXE, 441
/r switch in VBC.EXE, 46–47, 58
RAD (rapid application development) tools, 61–62
RaiseEvent keyword, 312–313
Raising events, 312–314
Rank property, 150
RCWs (runtime callable wrappers), 509–512,
515
Reachable objects, 396
Read-only behavior of strings, 125
Read-only properties, 198–199
Read/write properties, 198
Readability
colon-style syntax for, 207
field initializers for, 185
reference variables for, 180
ReadOnly fields, 174–177
ReadOnly keyword, 199
Recompiling
const fields, 175
optional parameters, 188, 194
Recordset objects, 511
ReDm statement, 137–138, 149
ReDm Preserve statement, 141–143
Redirection, version
disabling, 479–480
process of, 471–476
Redistributable .NET Framework, 5, 40
Reference counting, 394–396
in COM, 495–496
vs. garbage collection, 397–399
Reference list for dependent assemblies, 16
/reference switch in TLBIMP.EXE, 504–505
Referenced assemblies, locating, 47
References and reference types
with arrays, 136, 139, 144, 151
to assemblies, 55, 59, 438
assigning objects to, 73
circular, 398
comparing, 102–103
in CTS, 99–102
for dependent assemblies, 9, 59
with DirectCast, 118
exceptions with, 348
to external COM dependencies, 504
for fields, 173
interface-based, 259
Object class as, 113–114
passing, 187, 384–387
with polymorphism, 221
registering, 166
for shared members, 160
strings as, 121
in Visual Studio .NET, 58–61
Reflection for attributes, 11
Reflection namespace, 427
REGASM.EXE utility, 505–506, 527–529
Register for COM Interop option, 530
Registering
  event handlers, 307, 309–310, 314–328
libraries, 415, 527–528
primary interop assemblies, 503, 505
references, 166
Registry
  for applications, 419
  for CLSIDs, 528
  for COM DLLs, 490–491
  for primary interop assemblies, 505–506
  problems from, 414–415
  for strong name verification, 441
Registry Format for GUIDs, 521
Regression testing, 417
REGSVR32.EXE utility, 491
Release configurations, 66, 70
Release method, 496, 514–516
ReleaseComObject method, 515–516
Releasing COM objects, 514–516
remove_ prefix for event handlers, 309
Remove method, 310
RemoveHandler statement, 316
Removing
  assemblies from GAC, 450–451
  interface members, 261–264
Replacing methods, 224–225
chaining calls in, 229–231
design issues with, 230–232
dynamic binding in, 226–229
shadowing in, 237–245
static binding in, 225–226
Resolution, assembly, 16
Restarting AppDomains, 17
Resuming in exception handling, 343
Rethrowing exceptions, 344, 356
Return types and statements, 73, 187
  Finally blocks with, 368
  in method signatures, 186
  in overloading functions, 191
Reuse
  of base class implementations, 230
  of component libraries, 46
decoupling for, 259
inheritance for, 206
publisher policy files for, 476
Reverse-engineering, 49
Reverse method, 151–152
Revision numbers, 467, 469
  for assemblies, 427
  in version redirection, 472
Root namespace, 59–60
Rotor project, 7
Round method, 110
Rounding in conversions, 110
Running applications, 47
Runtime callable wrappers (RCWs), 509–512, 515
<runtime> element, 421
RunTimeVersion value, 528

S
Safe mode, 19–21
Safety checks with function pointers, 283
Satellite assemblies, 428
Scalability in DCOM, 35
Scope
  of AppDomains, 17
  modules for, 168
  namespaces for, 50
  for shared fields, 162
    as Visual Basic change, 74–76
Script interpreters, 497
Scripting clients in COM, 497–499
SDK (software developer’s kit), 40
Sealed classes, 207
Sealed structures, 388
Sealed value types, 373
Searching
  in arrays, 152
  for assembly files, 458
INDEX

Secondary AppDomains, 18–19
Secure Sockets Layer (SSL), 438
Security
  in AppDomains, 17
  CAS, 22, 456–457
  .NET Framework for, 34–35
  safe mode for, 21
Security assembly, 26
Security patches, 465, 468
Security risks, DLLs as, 415
Self-registering COM DLLs, 491
Semantic contracts with DLLs, 416
Semantics of methods, 186
sender parameter, 328–330, 332–333, 335
Server-side applications, 14
Server-side operating systems, 3
Set blocks, 196–199
Set methods with static event binding, 324–325
Set mutators, 196–197
Set statements in assignments, 73–74
Setup Wizard, 452–453
Shadowing methods and properties, 237–245
Shadows keyword, 238, 242–243
Shallow copies
  for arrays, 151
  for objects, 389–392
Shared constructors, 185–186
Shared events, 308
Shared keyword, 43, 96, 169
Shared members, 156
  combining with instance members, 166–168
  declaring, 172
  designing with, 163–166
  fields
    lifetime of, 162
    memory for, 161, 167
    scope for, 162
  vs. instance members, 96–97, 159–172
Shared methods, 43, 164–166
properties, 105, 198
SHFUSION.DLL file, 451
Short-circuiting logical operations, 86–87
Short type, 104, 374
Shutting down AppDomains, 17
Side effects, 242
Signatures
  for abstract members, 251
  for delegates, 284, 286, 328
digital. See Digital signatures
  in interfaces, 254
  method, 186–187
  in overloading, 190–192
  in overriding, 228
Simple Object Access Protocol (SOAP), 32, 36
Single-file assemblies, 8
Single inheritance
  with abstract classes, 253
  interfaces for, 253
Single-listener callback notifications, 277–278
Single-stepping, 67
Single type, 104, 106, 109
Single-rooted inheritance hierarchies, 95, 112, 210
Skeletons
  for event handlers, 322
  for interfaces, 257
SLN function, 28
SN.EXE utility, 431–432, 440–441, 448
SOAP (Simple Object Access Protocol), 32, 36
Software developer’s kit (SDK), 40
Solutions in Visual Studio .NET, 57
Sort method, 152
Sorting arrays, 152
Source files, editors for, 42
Source property for exceptions, 353, 356
Spoken languages, assembly settings for, 427–428
Square brackets ([ ]) for delegates, 295
SSCLI project, 7
SSL (Secure Sockets Layer), 438
StackTrace property, 353, 355–356
State management for performance, 35
Static binding
  of events, 318–327
  operation of, 224–226
StrCmp method, 130–131
StrComp function, 27–28
Strict type checking
  with cloning, 389
  with conversions, 81–84
  primitive type, 108–109
  string, 133
  with enumerations, 375–376
with interfaces, 260
with late binding, 518
with method overloading, 192
with Object class, 80, 112–113
with polymorphism, 222–223
as Visual Basic change, 78–85
String.Empty value, 124
StringBuilder class, 126–127
Strings and String class, 121–124
comparing
using equal signs, 130–132
using methods, 127–129
concatenating, 124–127
conversions with, 110–111, 132–133
formatting, 134–135, 430
members in, 96–97
vs. primitive types, 122–123
shallow copies with, 390–391
Strong names
in assemblies, 431–433
for GACs, 448
inspection for, 460
for interop assemblies, 501, 504
in private assembly deployment, 446
referring assemblies without, 438
verifying, 435–437, 441
Structure construct, 372–373, 379–380
Structured exception handling, 339
classes for, 346–350, 354–357
filtering, 350–352
Finally blocks in, 365–370
inner exceptions, 357–359
nesting in, 346
summary, 370
system-defined exceptions in, 359–362
throwing and catching exceptions, 339–345
user-defined exceptions in, 362–365
as Visual Basic change, 76–78
When clauses in, 352–353
Structures, 372
base types for, 210
vs. classes, 382–383, 387–388
creating, 379–382
for interface implementation, 380
object cloning in, 388–393
passing parameters in, 384–387
Sub keyword, 187
for delegates, 284
for event handlers, 310–311
Subclasses, 206
Superclasses, 206
SuppressFinalize method, 407–408
Symbolic debug logic, 66
System.Activator class, 517
System.Array class, 149–153
System assembly, 24–26
System.DateTime type, 107–108
System-defined exceptions, 348, 359–362
System.Enum class, 375–376
System.EventHandler delegate, 328–333
System.Exception class, 354–357
System.GC class, 402–403
System.Math class, 163
System.Object class. See Object class
System.Reflection namespace, 427
System.Runtime.InteropServices namespace, 512
System.String class, 121
System-supplied class libraries, 24–26
System.TimeSpan type, 107
System.ValueType class, 99–100, 372–373
System.Windows.Forms assembly, 26, 30, 62, 463
SystemException class, 348, 350
/t:exe switch in VBC.EXE, 43, 56
/t:library switch in VBC.EXE, 45, 58
tab constant, 107
Tampering, digital signatures for, 431–442
Target platforms, library support for, 23
TargetSite property, 353, 357
Templates, classes as, 156
Testing
applications, 67
configuration files, 421
DLLs for COM, 529–530
Text option for string comparisons, 130, 132
Text property for command buttons, 64
TextChanged event, 331–332
TextChangedHandler delegate, 332
Threads
in method execution, 97–99
in object finalization, 400–402
Throw statement, 77
StackTrace property for, 355–356
for user-defined exceptions, 363–364
Throwing exceptions, 76–77
INDEX

process of, 339–345
system-defined exceptions, 359–362
Time-sensitive resources, 404
Times in formatting code, 134
TimeSpan type, 107
.TLB extension, 486
TLBEXP.EXE utility, 520
TLBIMP.EXE utility, 500–502
Tolerance in AppDomains, 17
ToString method, 126, 132–134
in COM, 524
for enumerations, 375
ToUpper method, 28
Triggering garbage collection, 403
True value, 84–85
Try statement, 77, 341–345
cleanup code in, 366
for delegates, 302
for filtering exceptions, 350–352
nesting, 346
StackTrace property for, 356
for system-defined exceptions, 360–361
When clauses in, 352–353
Two-dimensional arrays, 148
Type libraries for COM DLLs, 486, 488–490
Type library exporter, 519–520
Type library importer, 500
TypeLoadException class, 349
TypeOf operator
with interfaces, 261
with polymorphism, 224
for sender parameters, 333
Types, 89
in arrays, 135, 143–146
in assemblies, 49
checking. See Strict type checking
COM vs. CLR, 487
CTS. See CTS (Common Type System)
cecapsulating, 157–159
in exception handling, 343
for fields, 172
filtering exceptions based on, 350–352
with function pointers, 283
nested, 202–203
with Object class, 120
in overloading functions, 191
with polymorphism, 218–224
for properties, 196
for return values, 187
tags for, 106
user-defined, 380

U
-u switch in GACUTIL, 450–451
UCase function, 27–28
UDTs (user-defined types), 380
Unboxing process, 115–118
Unicode characters
Char type for, 106–107
for strings, 121
Uniform resource identifiers (URIs), 454–455, 459
Unmanaged code, 20
Unreachable objects, 396
Unsafe mode, 20
Unsigned integers, 92
Up-casting with polymorphism, 222
Updating
Const fields, 175
DLL versions, 415–417
preexisting applications, 465
Upper bounds of arrays, 136, 138, 149
URLs (uniform resource identifiers), 454–455, 459
User-defined error codes, 494–495
User-defined exceptions, 348, 362–365
User-defined types (UDTs), 380
User-defined values, 372–373
enumerations, 373–379
structures, 379–388

V
/v switch in ALEXE, 478
Validation checks
for input parameters, 361
with wrapper methods, 28
Values and value types, 371
for arrays, 144
cleanup code for, 404
in CTS, 99–102
for fields, 173
passing, 187, 384–387
primitive types as, 104
summary, 411
user-defined, 372–373
enumerations, 373–379
structures, 379–388
ValueType class, 99–100, 372–373
Variables
in break mode, 67
declaring, 75, 79, 100–101
defining, 71
initializing, 71
memory for, 100–102
scoping rules for, 74
on stack frames, 98
Variant type, 118–119
.vb extension, 42
VBC.EXE compiler, 41–43, 56, 58
VbCrLf constant, 28
vbObjectError constant, 494
VBScript language, 498
VbTab constant, 28
Vendor boundaries, applications across, 35
Verification checks
for assembly producer identities, 431
for GACs, 448
for private assembly deployment, 446
for strong names, 435–437, 441
version attribute for <codeBase>, 454
Version numbers
for assemblies, 9, 16, 426–427
in compilation, 47
in format strings, 430
in versioning, 467–471
Version redirection
disabling, 479–480
process of, 471–476
Versions
for assemblies, 464
backward compatibility in, 464–467
CLR in, 469–471
publisher policy files in, 474–480
version numbers in, 467–469
version redirection in, 471–476
of base classes, 212–213
in code deployment, 34
of DLLs, 415–417
with GAC, 448
as migration challenge, 38
problems in
in COM, 524
with optional parameters, 194
Viruses
.NET Framework for, 34
safe mode for, 21
Visual Basic 6 projects, porting, 483–484
Visual Basic changes, 70–71
conveniences, 71–73
inconsistencies and idiosyncrasies, 73–74
logical comparisons vs. bitwise operations, 85–87
scoping rules, 74–76
strict type checking, 78–85
structured exception handling, 76–78
Visual Basic language, 12
Visual Studio .NET, 40–41
for configuration files, 422
development with, 56–58
for interfaces, 257
for interop assemblies, 502–503
as migration challenge, 37
references and namespaces in, 58–61
Visualizing interfaces, 261
-VI switch in SN.EXE, 441
VTables in COM, 489
-Vu switch in SN.EXE, 441
Web applications, 32, 35
Web assembly, 26
Web.config file, 420
Web Forms, 31
Web Service Description Language (WSDL), 36
Web services, 31–32
attributes for, 11
open standards for, 36
When clauses in Catch blocks, 352–353
Widening conversions, 83
Win32 API, 3–4
WINCV.EXE – The Windows Class Viewer, 55
Windows API, 2–3
Windows application, 62
collection files for, 422
creating, 62–66
Windows CE devices, 36
Windows Forms applications, 30–31
creating, 61–62
event handlers in, 329–330
Windows Registry
for applications, 419
for CLSIDs, 528
for COM DLLs, 490–491
for primary interop assemblies, 505–506
problems from, 414–415
for strong name verification, 441
INDEX

WithEvents fields
  in inherited classes, 327–328
  for static event binding, 318–326
Wrappers
  COM callable wrappers, 532–534
  for interop assemblies, 506–509
  runtime callable wrappers, 509–512, 515
Write-only properties, 198–199
WriteLine method, 68–69
WriteOnly keyword, 199
Writing software, 34
WSDL (Web Service Description Language), 36

X
  X property for mouse, 334
  XCOPY deployment, 34
    with GACs, 449
    for private assemblies, 443
  XML, 32, 36, 419–425
    XML assembly, 26, 58
    XML parser, 37
  XOR operator, 85
  XSD (XML Schema), 36

Y
  Y property for mouse, 334

Z
  Zero-based arrays, 135–136