



Index

- Abnormal termination, 199–200, 210–211, 379
- Abstract attribute, 162
- Abstract interfaces, 70
- Abstract methods, 165–172
 - declarations, 68–69
- Abstract types, 70, 78, 379
- Access modifiers
 - fields, 91–92
 - members, 50–51
- AcePowell class, 170
- Add method, 100, 175
- AddLimb method, 69
- AddPermission method, 312
- AddProc delegate type, 179
- age field, 71
- AL.EXE, 379
- All_Code group, 299
- AllMembershipCondition, 295
- AllowMultiple property, 111, 112
- Annotations, 357
- AOP (aspect-oriented programming), 3, 206, 379
- AOP-style interception, 207–213
- APPBASE directory, 34, 42
 - AppDomains, 266
- AppDomain policy level, 291, 292
- AppDomain type, 260
- AppDomain-Bound types, 275
- AppDomain.CreateDomain method, 257, 314
- AppDomain.CreateInstance method, 278–279, 371
- AppDomain.CurrentDomain, 258
- AppDomain.DoCallBack method, 258
- AppDomain.DynamicDirectory property, 267
- AppDomain.ExecuteAssembly method, 257–258
- AppDomain.RelativeSearchPath, 266
- AppDomains, 251–256, 379
 - APPBASE, 266
 - arbitrary code in, 258–259
 - assembly resolution and loading, 264–265
 - assembly resolver, 265–269
 - channel plumbing, 277
 - code management, 270–273
 - creating and destroying, 257
 - dynamic directories, 268
 - environmental properties, 257
 - events, 260–265
 - forcing to load and execute code, 257–258
 - hard threads, 254–256
 - identity table, 277–278
 - influencing JIT compiler, 270–272
 - objects, 252, 274–279
 - programming with, 256–259
 - resources owned by, 254
 - schedulable entity, 254
 - shadow copying, 268–269
 - soft threads, 254–256
 - terminal conditions, 260
 - types, 253
 - unhandled exceptions, 260
- AppDomainSetup type, 265, 266

- AppDomain.SetupInformation property, 265
- AppDomain.Unload method, 257
- ApplicationDirectory type, 283, 285
- ApplicationDirectoryMembership-Directory, 295
- application.INI file, 377
- Applications
 - component software, 205–206
 - contexts, 235–243
 - dynamically generating code, 266–268
 - factoring out pieces of, 206
 - probe path, 266
- Application-specific configuration files, 34, 41
- Applied .NET Framework Programming*, xx
- APP_NAME property, 268
- Array types, 135
- Array.IndexOf, 142
- Array.LastIndexOf, 142
- Arrays, 134
 - bulk copy operations, 141
 - common set of operations, 141
 - of context properties, 239
 - elements, 137
 - instantiating, 137
 - multidimensional, 138–140
 - nonzero lower bounds, 139
 - shape or capacity, 137–138
 - single-dimensional, 137
 - total number of elements, 136
 - type-compatible, 135
- Aspects, 3–4
- Assemblies, 379
 - actual path to load, 269
 - annotating for permissions, 321–322
 - applying certificates to, 288
 - changing version numbers, 45–46
 - culture-neutral, 25
 - defining, 17–23
 - delay signing, 29–30
 - delay-signing, 38
 - dependencies, 20–21
 - digital signatures, 27
 - display name, 25
 - encapsulation boundary, 21
 - explicitly loading, 84
 - finding appropriate, 40
 - hashes, 27
 - initializing, 31–38
 - loading alternative version, 33–34
 - loading by location, 31–32
 - location-independent names, 17
 - master list of externally referenced, 20–21
 - mixed-language, 17
 - multimodule, 17
 - multiple permission sets applied to, 296–297
 - multiple versions coexisting, 47
 - named and versioned as atomic unit, 18
 - on-demand loading, 31
 - physical paths, 17
 - public keys, 27–30
 - public keys as part of name, 286–287
 - public keys without valid signature, 29
 - publisher policies, 37–38
 - re-signing, 29
 - resolution and loading, 264–265
 - resource-only, 25
 - satellite, 25
 - scoping type definitions of component, 22–23
 - shadow copying, 269
 - signature mismatches, 46
 - signing, 28–30
 - to storing shared state, 47
 - types, 18
 - unique version numbers, 46
 - verifying signing, 30
 - version policies, 33–34
 - versioned as unit, 45
 - versioning hazards, 44–47
- Assembly evidence, 284
- Assembly linker, 37
- Assembly loader, 31–32, 379
- Assembly manifest, 18–21, 31
- Assembly names, 50
 - CultureInfo attributes, 24–25
 - four-part version number, 23–24
 - fully qualified reference, 25
 - loading by, 32–34
 - mapping to file name or URL, 41
 - Name property, 23
 - partially qualified reference, 25–27
 - public key, 25
- Assembly resolver, 32–33, 379
 - AppDomains, 265–269
 - CODEBASE hint, 41
 - CODEBASE URLs, 43

- configuration files, 34
- controlling behavior of, 265–269
- finding appropriate assembly, 40
- probing, 42–44
- setting properties, 266
- side-by-side installation of multiple
 - assembly versions, 47
 - version policies, 33–34
- assemblyBinding element, 34
- AssemblyLoad event, 264–265
- Assembly.Load method, 26, 27, 43, 83
- Assembly.LoadFrom method, 41, 83, 265
- Assembly.Location property, 269
- Assembly.QualifiedName property, 86
- Assembly.Resolve event, 265–266
- Asynchronous methods
 - completing execution, 192
 - completion routines, 194–195
 - concurrency, 195–197
 - invoking, 188–198, 258
 - messages as calls to, 207–213
 - pass-by-value parameters, 192
- AsynchState property, 191–192
- AsynchWait Handle property, 191
- AsyncProcessMessage method, 230
- Attribute property, 304
- auto attribute, 335

- Base base type, 170
- Base construct, 73
- base keyword, 70, 71
- Base types, 69–75, 171
 - constructors, 73–74
 - enumerating, 86
 - generic, 87
 - hiding methods, 72
 - non-private members, 70
 - private members, 89
 - runtime semantics, 70
- beforefieldinit attribute, 62, 104
- BeginInvoke method, 189, 190, 192
- Binary contract, 5
- BinarySearch method, 142
- BindingFlags.IgnoreCase flag, 90
- BindingFlags.NonPublic flag, 175
- BindToRuntimeHost method, 370–371
- BINPATH_PROBE_ONLY property, 268
- Blittable types, 350–351
- Boolean types, 114
- Boxing, 133–134

- Brown, Nat, xvii
- Built-in types, 114

- C#, 12
 - == and != comparison operators, 127–128
 - += and -= operators, 100
 - applying arbitrary attributes, 105
 - array creation, 136
 - base type constructor, 73–74
 - build-in types, 114
 - cast operator, 82
 - castclass opcode, 82
 - class definition, 116
 - constructors, 62
 - custom attributes, 106
 - declaring events, 101
 - delegate invocation, 184
 - delegate type definition, 180–181, 189–190
 - delimiter characters, 60
 - derived constructor, 73–74
 - destructor, 149
 - enum definitions, 117–118
 - final methods, 164
 - Finalize method, 149
 - hide-by-signature types, 72
 - implementing interface method, 167
 - indexed properties, 102
 - initializing arrays, 137
 - invoking delegate type constructor, 182–183
 - is keyword, 80, 81
 - isinst opcode, 80
 - as keyword, 80
 - literal field, 53–54
 - /main command-line switch, 17
 - Main method, 16
 - monitor-based locking, 198
 - new operator, 53, 121
 - operator overloading, 128
 - params keyword, 57
 - pinned pointers, 331–332
 - readonly keyword, 54
 - rectangular arrays initialization
 - syntaxes, 139
 - ref and out keywords, 58
 - ref or out keywords, 56, 123
 - scaled addition and subtraction, 333–334

- static keyword, 51
- struct definition base type, 116–117
- /target command-line switch, 14–15
- target of custom attribute, 106–107
- this keyword, 58
- try-finally construct, 148
- type definition, 56, 98–99
- type initializers, 60–61
- typeof operator, 82–83
- using statement, 151–152
- variable declaration, 138
- void keyword, 54
- C++, 12
 - == and != comparison operators, 127–128
 - applying arbitrary attributes, 105
 - constructors, 62
 - defining delegate types, 179–180
 - delimiter characters, 60
 - destructors, 148
 - hide-by-name types, 72
 - method table, 165
 - opaque types, 337–339
 - operator new, 121
 - operator overloading, 128
 - pinned pointers, 331–332
 - scaled addition and subtraction, 333–334
 - this keyword, 58
 - void keyword, 54
- C# compiler
 - beforefieldinit attribute, 62
 - new keyword, 71
- C++ compiler
 - /CLR switch, 15
 - CLR-compliant mode, 176
 - prohibiting inline assembly in managed methods, 177
- C# compiler (CSC.EXE), 14
- C++ compiler (CL.EXE), 14
- C# *Primer*, xx
- CACHE_BASE property, 269
- Caching native images, 154
- Call chains, 343–345
- Call context, 231–234
- calli instruction, 176
- callvirt opcode, 159
- CAs (certificate authorities), 287
- CAS (code-access security), 281–282
 - evidence, 283–290
- Case-insensitive search, 90
- caspol.exe, 291, 380
- castclass opcode, 79–80, 82, 165, 214, 215
- .cctor method, 61
- CCW (COM-callable wrapper), 355–357
- cert2spc.exe, 287
- Chaining
 - constructors, 63
 - delegates, 185–188
- Child code groups, 296
- Child domains, loading and executing code, 258
- CIL (Common Intermediate Language), 6, 380
 - converting to native machine code, 153–158
 - references to metadata, 7
 - translating into native machine code, 7
- CIL-to-native-translation, 8–9
- .class extern directive, 20
- Classes
 - exposing public fields, 220
 - implementing IComparable interface, 129
 - proxiable, 221
- Cleanup code, triggering execution, 148–152
- cl.exe, 380
- CLI (Common Language Infrastructure), 6, 380
- Client context sinks, 245–247
- Clone method, 130, 168
- Cloning objects, 130–132
- CloseHandle function, 349
- CLR (Common Language Runtime), xvii–xix, 6, 380
 - CIL-to-native-translation, 8–9
 - CodeDOM, 10–11
 - as COM component, 368–377
 - contracts between components, 6
 - execution scope, 251–256
 - flexibility, 6
 - GC (garbage collection), 146
 - loading, 361–368
 - managed execution, 9
 - metadata, 6–7
 - native code, 7
 - programming languages, 11, 12
 - programming model, 9–12

- referring to fields and methods by
 - name and signature, 6
 - resolving names to locations, 38–44
 - type definition, 6
 - verifiably type-safe code, 9
- CLR contracts, 6–7
- CLR event, 99
- CLR loader, xix, 31–38
- CLR modules
 - affiliated with only one assembly, 17
 - assemblies, 17–23
 - assembly manifest, 18–21
 - code, 13
 - CodeDOM, 14
 - compilers, 14–15
 - directly and indirectly loading, 18
 - .DLL file extension, 15
 - .EXE file extension, 15
 - file format, 14
 - main entry point, 16
 - metadata, 13–14
 - mscorlib assembly, 18
 - .netmodule file extension, 15
 - as opaque data, 13
 - PE/COFF executable file format, 13
 - resources, 14
 - /t:exe option, 15–16
 - /t:library option, 15
 - /t:module option, 15
 - translating source code into, 14–15
 - /t:winexe option, 15–16
 - as valid Win32 modules, 13
- CLR types, 50
 - assembly names, 50
 - definitions, 50–51
 - description, 50
 - identifying, 22
 - local name, 50
 - members, 50–51
 - names, 50
 - namespace prefix, 23, 50
- CLR-based .EXE programs, 34
- ClrCreateManagedInstance method, 371
- CLS (common language specification), 50, 380
- CMD.EXE, 363
- CoCreateInstance method, 360–361, 362, 369, 373
- Code
 - CIL (common intermediate language)
 - format, 13
 - domain-specific, 3
 - evidence about origins, 282
 - execution scope, 251–256
 - generation, 86–87
 - host evidence, 284
 - management and AppDomains, 270–273
 - non-verifiable, 282–283, 330
 - origins of, 283–290
 - verifiable, 282–283, 330
 - Code groups
 - altering interpretation, 297
 - built-in, 299
 - child, 296
 - finding matching, 297–298
 - hierarchy, 293–294
 - membership conditions, 294–295
 - root, 295
 - CODEBASE hints, 41, 47
 - CODEBASE URLs, 43, 284
 - Code-based URLs, 285
 - CodeDOM, 10–11, 14
 - CodeGroup type, 302
 - CodeGroup.Children property, 296
 - Codegroup.ResolveMatchingCodeGroups method, 297
 - code.netmodule module, 18, 20
 - CoGetObjectContext method, 368
 - Collect method, 146–147
 - COM (Component Object Model), 380
 - aspects, 3–4
 - binary contract, 5
 - complexity, 5
 - component contract, 5
 - component dependencies, 3
 - component versioning, 5
 - contracts, 1–4
 - IDL (Interface Definition Language), 3, 4
 - interface pointer, 5
 - lack of extensibility, 3
 - method table, 165
 - operation of contracts, 4
 - order-of-declaration problems, 6
 - physical contract, 5

- problems with, 2
 - as programming model, 2
 - as supporting platform technology, 2
 - TLB (Type Library) files, 3, 4
 - type definitions, 1, 2
 - type descriptions, 5
 - types, 2
- COM contracts, 5–6
- COM loader, 4
- COM programming model, 1–2
- COM+ Runtime, 6
- COM3, 6
- Combine method, 186–187
- COMException exception, 349–350
- CompareTo method, 128, 129, 130
- Compilers, 14–15
- CompletedSynchronously property, 190
- Completion routines, 194–195
- COMPLUS_DbgJITDebugLaunchSetting
 - environment variable, 261
- Component dependencies, 3
- Component software, 205–206
- Component versioning, 5, 7
- component.dll module, 18, 20
- Components
 - CIL (Common Intermediate Language), 7
 - contracts between, 2
 - developer, 25
 - identifying compilation of, 289–290
 - language and country code, 24–25
 - machine code, 6
 - security, 281–283
 - uniquely identifying developer, 27–30
- Concrete types, 68–69, 70
 - initializing, 166
- Concurrency, 195–197
- Concurrent Programming in Java*, 198
- Configuration files
 - application, 34, 41
 - ASP.NET applications, 34
 - assembly resolver, 34
 - CLR-based .EXE programs, 34
 - machine-wide, 34, 41
 - publisher policies, 37
 - publisherPolicy element, 38, 46
 - version policies, 34, 36–37
 - XML (Extensible Markup Language), 34
- Configuration root element, 34
- const keyword, 53
- const parameters, 351
- Constructors, 62–63, 380
 - base types, 73–74
 - chaining, 63
 - internal, 74
 - virtual method calls, 73
- Context attributes, 239–243
- Context properties, 238–239, 243
- Context wrapper, 4
- Context-agile objects, 235–239
- Context-bound objects, 235–239
- ContextBoundObject type, 223, 225–226
- Context.ContextProperties method, 239
- Context.GetProperty method, 239
- Contexts
 - context attributes, 239–240, 241–242, 243
 - context properties, 238–239, 240–241, 243
 - data associated with, 238
 - interception, 243–250
 - manipulating state of, 243
 - as property bag, 238
 - threads, 235
- Contracts, 1, 2, 51, 380
- ConvertAssemblyToTypeLib method, 357–358
- Copy method, 141
- COR (Component Object Runtime), 6
- CorBindToCurrentRuntime method, 371
- CorBindToRuntimeHost method, 369–371
- Cor_Enable_Profiling property, 374
- _CorExeMain, 366
- CORINFO_CLASS_STRUCT structure, 155, 166
- count field, 71
- cprdbg.exe, 380
- Create.ConstructionReturnMessage
 - method, 226
- created field, 54
- CreateDelegate method, 182
- CreateInstance method, 224–225
- Cross-AppDomain proxies, 276–278
- CrossAppDomainChannel, 277
- CrossAppDomainDelegate type, 258
- Cross-component contracts, 154
- CrossContextChannel type, 237–238
- Cross-mode invocation, 343
- Cross-process synchronization, 197
- CSC.EXE, 20, 358

- .ctor method, 63
- CTS (common type system), 49, 50, 380
- Culture property, 25
- Culture-neutral assemblies, 25
- Culture-neutral property, 25
- Current local variable, 56
- CurrentDomain property, 256
- Custom attributes, 104
 - inspecting fields at runtime, 110
 - parameters, 108–110
 - restricting application of, 111–112
 - stored as passive BLOBs, 107
 - strongly typed, 105
 - target, 106–107
- Custom proxies, 223–224, 226
- Deadlock, 195
- Debuggers, attaching, 261
- Declarative attributes, 4
- Declarative security demand, 315, 316, 317–318
- DeclaringType property, 87
- Deep copy, 131
- Default constructor, 62, 63
- Default policy files, 299
- Delay signing, 28, 29–30
- Delay-signed assemblies, 38
- Delegate objects
 - asynchronous method invocation, 188–198
 - delegate types, 178
- Delegate types, 178
 - instance constructor method, 179
 - invoking constructor, 182–183
 - members, 180
 - type name, 180
 - use of, 180
- Delegates, 178–188, 380
 - binding to static method and instance method, 181
 - chaining, 185–188
 - instantiating, 181
 - invoking, 183
- Demand method, 313
- Dependencies, 3
- dependentAssembly elements, 34
- Derived types, 70–71
 - ContextBoundObject type, 226
 - giving permission to, 322
 - rusing field name, 71
- Design Patterns*, 2
- Destructor, 148–149
- DEVPATH environment variable, 38–39
- Digital certificates, 27
- Digital signatures
 - assemblies, 27
 - overwriting, 30
- Direct base types, 80
- Display method, 167
- Display name, 25
- Dispose method, 149–151
- Distinguished methods, 60, 62
- .DLL file extension, 15
- DllGetClassObject method, 360, 366
- DllImport attribute, 355
- Dns.GetHostByName method, 313, 318
- DnsPermission permission, 317, 322
- DomainUnload event, 260
- Down-cast operator, 133
- Down-casting, 79
- Dynamic directories, 268
- Dynamic method dispatching, 69, 133
- Dynamic type coercion, 79–80
- DYNAMIC_BASE property, 268
- DynamicInvoke method, 183
- ECMA Web site, 6
- ecx register, 158, 160
- EJB (Enterprise Java Beans), 4, 235
- EJBObject.getHandle method, 235
- EMCA_Strong_Name code group, 299
- Encapsulation boundary, 21
- EndInvoke method, 189, 192, 194
- Enterprise policy level, 291, 292
- enterprisesec.config, 291
- EntryPoint parameter, 347
- Enumeration
 - member names, 118
 - [System.Flags] attribute, 119
- enumeration
 - types, 117–118
- EnvironmentPermission permission, 322
- Envoy sinks, 247–250
- Equals method, 124, 128
- Equivalence, 124–130
- ERROR_MOD_NOT_FOUND error code, 349
- Essential COM*, 1, 2
- Event handlers, 99, 380
- EventInfo object, 99

392 ■ INDEX

- Events, 380
 - AppDomains, 260–265
 - event handlers, 99
 - methods, 100
- Everything permission set, 301
- Evidence, 381
 - assembly evidence, 284
 - based on certificates, 287–289
 - based on strong name, 286–287
 - dynamically producing permission sets, 302
 - host evidence, 284
 - identity permissions, 304
 - permissions based on, 292
 - programmatic types, 285–286
 - running through security policy, 302–303
 - XML-based representation of, 290
- Evidence type, 289
- ExactSpelling parameter, 347
- Exception handlers, 200–203, 381
- Exception object, 199–200
- Exception property, 211
- Exception table, 200–203
- Exceptions, 260–261
- exceptions, 381
- Exclusive locking, 197
- .EXE file extension, 15
- ExecuteMessage method, 217–218
- Execution scope, 251–256
- Executive, 206
- Explicit attribute, 335–336, 337
- Explicit method invocation, 172–178
- Explicit type initializer method, 61, 62
- Extensibility of metadata, 104–112
- External DLLs, errors raised by, 348

- Factory methods, 222–223
- famandassem access modifier, 22
- __fastcall stack discipline, 158
- FieldGetter method, 220
- FieldInfo type, 91
- Fields, 51–52, 381
 - access modifiers, 91–92
 - constant value, 53–54
 - declared type, 91
 - heap-allocated instances, 51
 - initonly attribute, 54
 - instances, 91
 - non-static, 51
 - reordering, 53
 - runtime or compile-time constant
 - expression, 91
 - static, 51, 91
 - value types, 114
- FieldSetter method, 220
- File-based URLs, 286
- FileCodeGroup code group, 302
- FileIOPermission permission, 302, 306, 310, 322
- final attribute, 162
- Final methods, 164, 170
- Finalization, 381
- Finalize method, 149
- Fire-and forget invocation, 194
- FlattenHierarchy flag, 89
- Forking stream of execution, 188–189
- Formatted types, 335
- FreeNamedDataSlot method, 232
- FullName property, 86
- Full-trust assemblies list, 294
- FullTrust permission set, 299, 301
- Fully qualified reference, 25
- Function pointers, 176
- Functions returning HRESULTs, 349–350
- FUSION API (application programming interface), 40–41
- FUSION.DLL component, 40
- fuslogvw.exe, 381

- GAC (global assembly cache), 37, 40–41, 47, 381
- GACUTIL.EXE, 40, 381
- Garbage collector
 - exposing, 146–147
 - identifying object as unreachable, 148
 - usage, 364–365
- GC (garbage collection), 143, 146, 381
- __gc modifier, 338
- GCHandle.Alloc method, 339
- GCHandle.Free method, 339
- gcroot.h file, 339
- Generative programming, 10
- Generic base type, 87
- GetAddMethod method, 99
- GetCurrentDomainData method, 272
- GetCustomAttributes method, 107, 110, 112
- GetData method, 232, 257, 259, 266, 274–279

- GetDefaultDomain method, 369
- GetFields method, 91
- GetFunctionPointer method, 175–176
- GetHashCode method, 128
- gethostbyname API function, 319
- GetLastError method, 348
- GetLength method, 139
- GetMembers method, 87, 90, 91
- GetProcAddress method, 346, 347
- GetPropertiesForNewContext method, 240, 242, 250
- GetProxiedType method, 215
- GetRealProxy method, 214
- GetRemoveMethod method, 99
- GetServerContextSink method, 247
- GetSetMethod method, 97
- GetType method, 83, 84
- GetTypeFromHandle method, 82, 84
- Global exception handler, 260–261

- Handle, 78–79
- Hard threads, 254–256, 381
- Hash type, 283, 289–290
- Hashtable object, 145, 146
- Heap, allocating new instance on, 53
- Heap-allocated instances fields, 51
- Hide-by-name methods, 71–72
- hidebysig attribute, 72–73, 104
- Hide-by-signature methods, 71–72
- Homogeneous, same-mode invocation, 343
- Host evidence, 284
- HRESULT, 349–350, 357
- HTTP-based URLs, 285

- IAAsyncResult interface, 190, 192
- IAAsyncResult.AsyncState property, 194
- IAAsyncResult.IsCompleted property, 191
- IBillee type, 66, 67
- ICalculator interface, 215
- ICloneable interface, 168
- ICloneable.Clone method, 168
- IComparable interface, 129, 142
- IConstructionCallMessage interface, 226
- IConstructionReturnMessage interface, 226
- IContextAttribute interface, 223, 239
- IContributeClientContextSink, 245
- IContributeEnvoySink interface, 247
- IContributeObjectSink interface, 245
- IContributeServerContextSink interface, 245
- ICorDebug interface, 373
- ICorDebugManagedCallback interface, 373
- ICorProfilerCallback interface, 374
- ICorRuntimeHost interface, 368
- IcorRuntimeHost.GetDefaultDomain method, 371
- ICorThreadPool interface, 371
- ICorThreadpool::CorSetMaxThreads method, 193
- ICustomAttributeProvider interface, 107, 110
- ICustomMarshaler interface, 351
- IDebuggerThreadControl interface, 371
- Identity, 124–130
- Identity permissions, 304, 324
- IDisposable.Dispose method, 150
- IDL (Interface Definition Language), 3, 4, 381
- IDrawable interface, 167
- IDrawable.Display method, 167
- IETF (Internet Engineering Task Force), 24
- If-else statement, 130
- IGCHost interface, 371
- IGCThreadControl interface, 371
- IIdentityPermissionFactory interface, 304
- ILASM method, 12, 177, 180
- ILASM.EXE, 381
- ILDASM.EXE, 238, 381
- ILogicalThreadAffinative interface, 234, 248
- IMessage interface, 207
- IMessageSink interface, 228
- IMetaDataConverter interface, 371
- IMetaDataEmit interface, 14, 87
- IMetaDataImport interface, 104
- IMethodMessage interface, 207–210
- IMethodMessage.GetArg method, 210
- IMethodReturnMessage interface, 210–213
- Immutable objects, 114, 128
- Imperative security demand, 315–316, 318
- Implementations, 6
- Indexed properties, 102
- Indexer, 102
- IndexOf method, 142
- Indirect base types, 80

- Indirect method invocation, 178–188
- Inheritance and interfaces, 66–68
- Inherited parameter, 107
- Inherited property, 111
- Initialization of types, 60–63
- InitializeServerObject method, 226
- initonly attribute, 54, 104
- instance constructor method, 180
- Instance flag, 90
- Instances
 - affiliation with types, 77
 - fields, 92
 - primitive types, 77
 - reference types, 119
 - user-defined types, 77
 - value types, 119
- Interception, 4, 206, 243–250
- interface attribute, 64
- Interface offset table, 166
- Interface pointer, 5
- Interface table, 80
- Interface types, 64, 80
- Interfaces, 165–172
 - abstract, 70
 - abstract method declarations, 68–69
 - annotations, 357
 - describing, 166
 - enumerating, 85–86
 - final methods, 170
 - implying virtual method dispatch, 219
 - importance of, 64–65
 - inheritance, 66–68
 - interface type declaring compatibility
 - with, 66–67
 - members, 64
 - method table slots, 166
 - name collisions, 169–170
 - nonfinal method, 170
 - proxy-friendly, 219
 - type compatibility with multiple, 66
 - type names, 64
 - types, 64–69
- Internal constructors, 74
- Internal members, 21, 22
- Internal types, 21, 22
- Internet permission set, 301–302
- Intersect method, 306, 307
- Invocation
 - cross-mode, 343
 - fire-and forget, 194
 - homogeneous, same-mode, 343
 - one-way, 194
- Invoke method, 180–190, 217, 226
- Invoking delegates, 183
- IPermission interface, 312–313
- IRemotingTypeInfo.CanCastTo method, 216
- IsAssignableFrom method, 84
- IsCompatible method, 85
- IsContextOK method, 239–240, 242
- IsDefined method, 107–108
- ISelfFundedPatient interface, 66–67
- isinst opcode, 79–80, 82, 165, 214, 215
- IsJITOptimizerDisabled attribute, 376–377
- IsJITTrackingEnabled attribute, 375–376
- IsRelatedType method, 85
- IsSubclassof method, 84–85
- IsSubsetOf method, 306, 307
- IStackWalk.Assert method, 346
- Item property, 103
- IValidator interface, 371

- Jagged or Java-style array, 140–141
- JIT (just-in-time) compiler
 - AppDomains influencing, 270–272
 - castclass opcode, 80
 - __fastcall stack discipline, 158
 - inlining method call through interface-based reference, 219
 - isinst opcode, 80
 - jmp instruction, 156–158
 - loading, 31
 - method table, 155–156
 - methods, 153–158
 - stub routines, 156
 - suppressing inlining, 219–221
 - type definitions, 31
- JIT debugger, 261
- JIT_ChkCast function, 80
- JIT_IsInstanceOf function, 80
- jmp instruction, 156–158
- JNI thunking layer, 328
- jsc.exe, 381
- JScript, 12
- JVM (Java virtual machine), 10

- Kiczales, Gregor, 206
- Kindel, Charlie, xvii
- Kirtland, Mary, xvii

- ldftn opcode, 183
- ldvirtftn opcode, 183
- Length property, 136, 139, 140
- Lightning, 6
- Load method, 32–33
- Loader, 381
- Loader optimizations, 271–272
- LoadFrom method, 31, 32
- Loading CLR, 361–368
- LoadLibrary method, 13, 346, 347, 361, 362
- Local Intranet URLs, 285
- Local name, 50
- Local variables, 51, 119
- LocalIntranet permission set, 301–302
- Lock contention, 196
- Locking resources, 195–197
- LogicalCallContext property, 232

- Machine policy level, 291, 292, 299, 301
- machine.config file, 34, 38–39
- Machine-wide configuration files, 34, 41
- Main method, 16–17
- makecert.exe, 287
- Malicious sites, 285
- Managed execution, 9
- Managed execution mode, 143, 341
- Managed methods, 341, 343
- Managed pointers, 329
- Managed signatures, 350
- Marshal method, 274
- MarshalAs attribute, 355
- Marshal.BindtoMoniker method, 361
- MarshalByRefObject type, 223
- Marshal.ChangeWrapperHandleStrength method, 356
- Marshaling, 274–279
 - by reference, 275
 - by value, 275–276
- Marshal.ReleaseComObject method, 356
- MemberInfo type, 87, 90
- Members, 50–51
 - access modifiers, 50–51
 - controlling access to, 51
 - delegate types, 180
 - enumerating, 87–91
 - enumeration types, 117
 - explicit qualification, 58
 - fields, 51–52
 - interfaces, 64
 - internal, 21, 22
 - methods, 51
 - name as string, 87
 - nested types, 51, 59–60
 - packaging common, 70
 - per-instance, 51
 - per-type, 51
 - public, 21
 - runtime description, 87
- Membership condition types, 295–296
- Membership conditions, 294–295
- MemberType property, 87
- MemberTypes flag, 90
- MemberwiseClone method, 130, 132
- Memory, 327–340
 - allocation, 51
 - Memory buffers, direct access, 327
 - Memory reclamation, 143
- Message sinks, 228–230, 237, 243–245
 - communicating with each other, 231
 - factory for, 243
- Messages
 - call context, 231–234
 - as method calls, 207–213
 - processing, 227–234
 - transitions and stack, 213–218
- Metadata, XVIII, 6–7, 50, 381
 - attribute field, 104
 - CLR modules, 13–14
 - component dependency, 7
 - extensibility, 104–112
 - extensible with custom attributes, 6–7
 - generative programming, 10
 - methods, 172
 - programming, 86–96
 - properties, 96
 - version information, 7
- metadata attribute, 104
- Method invocations, 157–165, 217
- Method pointer, 178
- Method table offsets, 160
- Method tables
 - determining which to use, 159
 - indexing, 160
 - interface offset table, 166
 - non-virtual methods, 159
 - slots, 160, 162
 - specific slot in, 158
 - types, 157
 - virtual methods, 159–160

- MethodBase property, 209
- MethodBase.GetFunctionPointer method, 177
- MethodInfo type, 93–95, 99, 112, 175
- MethodInfo.Invoke method, 172–173
- methodoffset, 160
- Methods, 51, 54–56, 381
 - abnormal termination, 199–200, 210–211
 - asynchronous invocation, 188–198
 - binding on objects, 178–188
 - calculating permissions, 313
 - circumventing access modifier, 175
 - class-based users, 170
 - events, 100
 - exception table, 200–203
 - explicit invocation, 172–178
 - explicit overloads, 65
 - factory, 222–223
 - final, 164
 - hide-by-name, 71–72
 - hidebysig attribute, 72–73
 - hide-by-signature, 71–72
 - indirect invocation, 178–188
 - interface-based uses, 170
 - invocation through interface-based reference, 166
 - JIT (just-in-time) compilation, 153–158
 - local variables, 51
 - managed, 341, 343
 - managed signatures, 350
 - messages as calls to, 207–213
 - metadata, 172
 - multiple definitions, 58
 - nested types, 60
 - nonpublic, 175
 - non-static, 54
 - normal termination, 199, 210
 - overloading, 58
 - parameters, 55–57
 - pass-by-reference parameters, 122, 123, 329
 - passing parameters by value, 122–123
 - permissions, 313–314
 - properties, 96–97
 - registering or revoking event handlers, 99–101
 - restricting access to, 54–55
 - returning no value, 54
 - returning typed value, 54
 - special, 96–104
 - special-case handling, 226–227
 - stack frame, 173
 - static, 54, 55
 - stub routine, 155
 - supporting pre- and postprocessing calls, 221
 - termination, 198–204
 - types, 157–165
 - type-safe version, 167, 169
 - unmanaged, 341–343
 - unrestricted access to members, 57–58
 - value types, 114
 - virtual, 159, 224
 - visible to corresponding interface type, 167
- Mixed-language assemblies, 17
- Modes of execution, 340–345
- Modules, 382
 - assembly manifest, 31
 - initializing, 31–38
 - loading, 31–38
 - metadata, 50
 - MVID (module version identifier), 154
 - on-demand loading, 31
 - unmanaged, 345–361
- Monitor, 197
- Monitor-based locking, 197–198
- mov instructions, 79, 160
- mscorlib.dll library, 373
- MSCOREE.DLL library, 273, 362–368
- mscorlib.tlb, 369
- mscorlib assembly, 18, 23, 238, 299
- mscorlib.tlb, 369
- MSCORPE.DLL library, 153–154
- MSCORSVR.DLL library, 362–363
- MSCORWKS.DLL library, 362
- MTS (Microsoft Transaction Server), 3, 207, 382
 - executive, 206
 - interception, 206
 - protecting this reference, 235
- MTS attributes, 4
- MTS executive, 4
- Multidimensional arrays
 - jagged or Java-style array, 140–141
 - rectangular or C-style, 138
- MultiDomain flag, 271, 272, 273
- MultiDomainHost flag, 272–273
- Multimodule assemblies, 17
- MVID (module version identifier), 154

- Mycomputer zone, 285
- My_Computer_Zone code group, 299
- Name collisions, 71
- Name property, 86, 87, 286
- Named slots, associating data with, 238
- namespace prefix, 23, 50
- Nested types, 51, 59–60
- .NET Framework, xvii–xix
- .NET framework, 379
- .NET Framework SDK (software development kit), 11, 379
- .NET XML stack, 23
- NetCodeGroup code group, 302
- .netmodule file extension, 15
- new operator, 53, 132, 138
- newobj opcode, 121, 223, 224, 225, 239, 250
- newslot attribute, 160, 171
- NextSink property, 228
- NGEN.EXE, 153–154
- ngen.exe, 382
- Nonblittable strings, 354
- Nonblittable types, 350–351
- Non-interface types, 70
- Non-opaque types, 338
- NonPublic flag, 89
- Nonpublic methods, 175
- Nonrepudiation, 27
- Nonroot references, 143
- nonsidebysideappdomain attribute, 48
- nonsidebysidemachine attribute, 48
- nonsidebysideprocess attribute, 48
- Non-static fields, 51
- Non-static methods, 54
- Non-verifiable code, 282–283, 330
- Non-virtual methods, 159
- Normal termination, 199, 210, 382
- NormalException exception, 200
- Nothing permission set, 299
- Numeric types, 114
- Object activation, 223
- Object handles, 278–279
- Object headers, 78–79
- Object references, 147, 328
- Object serialization, 275
- Object sinks, 245
- Object.Equals method, 126–127, 129
- Object.Finalize method, 148
- Object.GetHashCode method, 127
- Object.MemberwiseClone method, 130
- Objects, 113–119
 - acting like values, 128
 - affiliation with types, 78
 - AppDomains, 252, 274–279
 - associating locks with, 197
 - associating resources with, 78
 - binding methods on, 178–188
 - cloning, 130–132
 - context-agile, 235–239
 - context-bound, 235–239
 - contexts, 234–243
 - deep copy, 131
 - definition of, 114
 - equivalence, 124–130
 - finalization, 148–152
 - identical, 124
 - identity, 124–130
 - immutable, 114, 128
 - implementing specific features, 245
 - life cycle, 143–147
 - nonroot reference, 143
 - object header, 78–79
 - polymorphism, 114
 - preventing from being garbage-collected, 143–145
 - reclaiming, 146–147, 148
 - root reference, 143
 - shallow copy, 130–131
 - state, 54
 - subsets, 66
 - sync blocks, 197
 - tracking references, 143
 - type handle, 165
 - valid, 119
 - weak references, 145–146
- One-way invocation, 194
- OnSubmit event, 100, 101
- Opaque types, 337–339
- Overloading methods, 58
- Overloads keyword, 72
- override keyword, 164, 170
- Overriding virtual methods, 164–165
- Packaging common members, 70
- ParamArrayAttribute attribute, 57
- Parameters, 55–57
 - custom attributes, 108–110
 - pass-by-reference, 56

- pass-by-value, 56
 - reference type, 123
 - value types, 122–123
- Partially qualified reference, 25–27
- partialName attribute, 27
- Pass-by-reference parameters, 56
- Pass-by-value parameters, 56
- Patient type, 168
- PE/COFF executable file format, 13
- Peitrek, Matt, 156
- Per application version policies, 36–37
- Per component version policies, 36–37
- Per machine version policies, 36–37
- perinstance members, 51
- Permission sets, 282, 293, 305, 382
 - built-in, 299–302
 - dynamically producing, 302
 - multiple applied to assembly, 296–297
 - programmatically exposing, 304
 - top-of-stack, 314
- Permissions, 282, 304–312, 382
 - annotating assembly for, 321–322
 - asserting, 320–321
 - based on evidence, 292
 - built-in, 308
 - built-in categories, 304–305
 - calculating for methods, 313
 - combining, 308–310
 - derived types and, 322
 - identity, 324
 - methods, 313–314
 - performing set operations on, 310–312
 - permission sets, 305
 - restricting, 317–319
 - states, 307
 - trusted components, 319–320
 - type not satisfied, 314
 - types, 306–307
 - union of, 303
 - unrestricted, 307–308
- PermissionSet type, 308, 310–312, 313
- PermissionState.Unrestricted, 312
- @perSYSTEM_ROOT@per directory, 363
- Per-type members, 51
- pete.netmodule subordinate module, 20
- Physical contract, 5
- Pinned pointers, 331–333
- P/Invoke, 345–361
- Pointers, 328
 - managed, 329
 - pinned, 331–332
 - unmanaged, 329–331
- Policy, 282
- Policy files, 299
- Policy hierarchy, 292–293
- Policy levels
 - code group hierarchy, 293–294
 - full-trust assemblies list, 294
 - permission sets, 293
- PolicyLevel type, 302
- PolicyLevel.ResolveMatchingPolicyGroupps method, 298
- PolicyStatementAttribute attribute, 302
- PolicyStatementAttribute.Exclusive property, 297
- PolicyStatementAttribute.LevelFinal property, 297
- Polymorphic types, 133
- Polymorphism, 114
- Poor scalability, 196
- #pragma managed directive, 341
- #pragma unmanaged directive, 341
- Precompiling, 153
- Predicates, 200–203
- PreserveSig parameter, 349
- PreStubWorker routine, 156
- Primary interop assembly, 360
- Primitive types, 77, 114, 382
- PriorityProxy.Invoke method, 230
- Private key, 27–28, 29
- Private members, 89
- Private probe paths, 47
- Private types, 21
- Probe path, 266
- Probing, 42–44
- Probing element, 266, 268
- Processes, 251, 271–273
- ProcessExit event, 260
- Processing messages, 227–234
- ProcessMessage routine, 217
- Program verification, 328
- Programming
 - AppDomains, 256–259
 - metadata, 86–96
- Programming languages
 - abstract attribute, 162
 - accessing array elements, 136
 - applying arbitrary attributes, 105
 - array types, 135
 - built-in types, 114

- CLR, 11, 12
 - declaring array variables, 136
 - defining events as members of type, 101
 - defining indexer, 102–104
 - exposing indexers, 102
 - final attribute, 162
 - indexer accessed explicitly by name, 103
 - initializing arrays, 136
 - news slot attribute, 162
 - obfuscating semantics, 98
 - primitive types, 114
 - registering or revoking event handlers, 100–101
 - syntax for defining properties as members of type, 98–99
 - virtual attribute, 162
- Programming model, 9–12
- Programs
 - CLR modules, 13–17
 - concurrency, 195–197
 - forking stream of execution, 188–189
- Properties, 96–98, 102, 382
- PropertyInfo.Name member, 97
- PropertyInfo.PropertyType member, 97
- Protected types, 21–22
- Proxiable classes, 221
- Proxiable types, 219–227
- Proxies
 - CCW (COM-callable wrapper), 355–357
 - custom, 223–224, 226
 - priority refactored as message sink, 228–230
 - remotes, 275
- ProxyAttribute type, 223–224
- Pseudo-custom attribute, 105
- Public flag, 89
- Public key token, 27, 382
- Public keys, 25, 382
 - assemblies, 27–30
 - calculating token, 27
 - location of, 29
- Public members, 21
- Public types, 20, 21
- PublicKey property, 286
- PublicKeyToken property, 25
- Public-only key files, 27
- Public/private files, 27–28
- Publisher policies, 37–38
- Publisher policy files, 46
- Publisher type, 283, 287–289
- publisherPolicy element, 38
- qualifyAssembly element, 40
- RCW (runtime-callable wrapper), 355–357
- ReaderWriterLock, 197
- ReadLine method, 143
- readonly keyword, 54
- RealProxy constructor, 215, 219
- RealProxy.GetTransparentProxy method, 214
- Recalc method, 56
- Reference type variables, 119, 120
- Reference types, 113, 114, 121, 382
 - arrays, 134, 135, 137
 - assignment, 121
 - default field value, 119
 - default implementation of Object.Equals, 127
 - instances, 119
 - parameters, 123
 - synthesized, 134
- ReflectedType property, 87
- Reflection
 - code generation, 86–87
 - System.Reflection.BindingFlags parameter, 89–90
- REGASM.EXE, 360, 366, 382
- regsvcs.exe, 382
- Relative search path, 266
- Remote-unaware, 275
- RemotingServices.ExecuteMessage method, 217–218, 228, 230
- RemotingServices.IsTransparentProxy method, 214
- Remove method, 100, 186–187
- Request messages, 210, 212
- resgen.exe, 382
- Resolve method, 302, 303
- Resolving names to locations, 38–44
- Resource-only assemblies, 25
- ResourceResolve event, 265
- Resources
 - associating with objects, 78
 - CLR modules, 14
 - initializing, 31–38
 - loading, 31–38
 - locking, 195–196
 - owned by AppDomains, 254

400 ■ INDEX

- Response messages, 210
- ret instruction, 199
- ReturnValue property, 210–211
- rgn variable, 137
- Richter, Jeff, 156
- Root code group, 295
- Root references, 143
- RP (real proxy), 213–217
- RSA public/private keys, 27–28
- Runtime, 143
- Runtime types, 77–86
- Runtime.Remoting.RemotingServices type, 274
- RuntimeTypeHandle, 80

- SafeRef API, 235
- Satellite assemblies, 25
- Scoped method names, 168
- SCPs (Software Publisher Certificates), 288
- sealed keyword, 164
- Sealed types, 70, 74, 382
- Security
 - components, 281–283
 - declarative demand, 315, 316, 317–318
 - enforcement, 312–325
 - forcing security check, 322–324
 - imperative demand, 315–316, 318
 - permission sets, 282
 - permissions, 282, 304–312
 - security policies, 282, 291–303
- Security policies, 282, 382
 - enforcing, 312–325
 - policy hierarchy, 292–293
 - running evidence through, 302–303
- SecurityAction.Deny, 317
- SecurityAction.InheritanceDemand attribute, 322, 324–325
- SecurityAction.LinkDemand attribute, 322–324
- SecurityAction.PermitOnly, 317
- SecurityAction.RequestMinimum attribute, 321
- SecurityAction.RequestOptional attribute, 321
- SecurityAction.RequestRefuse attribute, 321
- Security.Demand attribute, 323
- SecurityException.PermissionType property, 314
- SecurityManager.ResolvePolicy method, 303–304
- SecurityManager.ResolvePolicyGroups method, 298
- SecurityPermission type, 306, 310
- SecurityPermissionFlags.Assertion permission, 319
- SecurityPermissionFlag.SkipVerification permission, 330
- SecurityPermissionFlags.Unmanaged-Code permission, 319
- Security.Permissions.PermissionState enumeration, 307–308
- Security.Zone.Trusted type, 285
- Security.Zone.Untrusted type, 285
- sequential attribute, 335
- Server context sinks, 245–247
- SetData method, 232–233, 257, 259, 266, 274–279
- SetLastError method, 348, 349
- SHA (Secure Hash Algorithm) hasing, 27
- Shadow copying, 268–269, 383
- SHADOW_COPY_DIRS property, 269
- Shadows keyword, 72
- Shallow copy, 130–131
- Shared locking, 197
- Shared types, 47
- SHFUSION.DLL, 41
- Side-casting, 79
- Signature mismatches, 46
- signcode.exe, 288
- Single-dimensional arrays, 137
- SingleDomain flag, 271, 272
- Single-method interface, 178
- Site type, 283, 284, 286
- Size type, 121
- SkipVerification permission, 330
- Sleep method, 347
- Slots, 160, 162
- SN.EXE, 27–30, 383
- .SNK file extension, 27
- soapsuds.exe, 383
- Soft threads, 254–256, 383
- Source code, translating to CRL modules, 14
- Special methods, 96–104
- Special-case categories of URLs, 285–286
- specialname attribute, 100
- Specifications, 6
- Stack and message transitions, 213–218

- Stack builder sink, 217–218
- Stack frames, 173
 - chains, 343–345
 - exception table, 200–203
- Start method, 170, 369
- Static field declarations, 61
- Static fields, 51, 91
- Static flag, 89
- Static member names, 58
- Static methods, 54, 55
- Stop method, 170
- String types, 114
- StringBuilder class, 354–355
- Strings, nonblittable, 354
- Strong names, 383
- StrongName type, 283, 286–287
- Stutz, David, xvii
- Subordinates, loading on demand, 31
- Subtract method, 177
- switch statement, 213
- Sync block index, 78
- Sync blocks, 197
- SynchProcessMethod method, 231–232
- SyncProcessMessage method, 228, 230, 250
- SysAllocString method, 353
- System assembly, 23
- System.AppDomain type, 256
- System.AppDomain.SetAppDomainPolicy method, 291
- System.ApplicationException exception, 200
- System.ArgumentException exception, 173, 286
- System.Array type, 134–135, 142
- System.AsynchCallback delegate, 194
- System.Attribute type, 105
- System.AttributeUsageAttribute attribute, 111
- System.BadImageFormatException exception, 31
- System.Byte type, 118
- System.Char array, 353
- System.CLSCompliant attribute, 50
- System.CodeDOM, 87
- System.Collections.ICollection interface, 289, 304
- System.ContextBoundObject method, 220
- System.Decimal type, 97, 99
- System.Delegate type, 178, 181, 183
- System.Delegate.Combine method, 101
- System.Delegate.Remove method, 101
- System.Diagnostics.DebuggableAttribute attribute, 375
- System.EnterpriseServices.ContextUtil type, 368
- System.EnterpriseServices.ServicedComponent type, 367
- System.Enum type, 117
- System.EventHandler type, 100
- System.Exception exception, 199–200, 213
- System.FileNotFoundException exception, 31
- System.GC class, 146
- System.GC.Collect method, 333
- System.GC.KeepAlive method, 147
- System.GC.SuppressFinalize method, 150
- System.ICloneable interface, 130, 167, 169
- System.IComparable interface, 128–129
- System.IDisposable interface, 149–150
- System.Int32 type, 180, 196
- System.IntPtr type, 179, 333–334
- System.IntPtr.Size property, 334
- System.InvalidCastException exception, 80
- System.LoaderOptimization optimization flag, 271, 370
- System.LoaderOptimizationAttribute attribute, 273
- System.MarshalByRefObject type, 219–220, 275
- System.MulticastDelegate type, 178, 185–186
- System.Net.Dns class, 315
- System.Net.DnsPermission permission, 313
- System.NullReferenceException exception, 119, 159
- System.Object type, 18, 23, 65, 70, 84, 124, 133, 179, 202, 274
- System.Object.Equals method, 129
- System.Object.GetType method, 84, 85
- System.Object.ReferenceEquals method, 128
- System.Object.ReferenceEquals static method, 124
- System.Reflection.AssemblyKeyFile attribute, 29
- System.Reflection namespace, 87

402 ■ INDEX

- System.Reflection.Assembly class, 31, 32, 265
- System.Reflection.AssemblyCulture attribute, 24
- System.Reflection.AssemblyDelaySign attribute, 29
- System.Reflection.Assembly.Evidence property, 290
- System.Reflection.Assembly.GetName method, 23
- System.Reflection.AssemblyName type, 23
- System.Reflection.AssemblyVersion attribute, 23
- System.Reflection.BindingFlags parameter, 89–90
- System.Reflection.ConstructorInfo type, 93
- System.Reflection.Emit namespace, 14, 87
- System.Reflection.FieldInfo type, 93
- System.Reflection.MethodBase type, 93
- System.Reflection.MethodInfo type, 93, 172
- System.Reflection.TargetException exception, 172
- System.Reflection.TargetParameterCount Exception exception, 173
- System.Runtime.InteropServices namespace, 357
- System.Runtime.InteropServices.CoImport attribute, 360–361
- System.Runtime.InteropServices.DllImport attribute, 346
- System.Runtime.InteropServices.FieldServices attribute, 335
- System.Runtime.InteropServices.GCHandle type, 338–339
- System.Runtime.InteropServices.Guid attribute, 361
- System.Runtime.InteropServices.MarshalAs attribute, 351
- System.Runtime.InteropServices.Marshal.GetLastWin32Error method, 348
- System.Runtime.InteropServices.PreserveSig attribute, 357
- System.Runtime.InteropServices.RuntimeEnvironment.GetRuntimeDirectory method, 364
- System.Runtime.InteropServices.RuntimeEnvironment.GetSystemVersion method, 364
- System.Runtime.InteropServices.StructLayout attribute, 335
- System.Runtime.InteropServices.TypeLibConverter attribute, 357
- System.Runtime.InteropServices.RuntimeMethodHandle type, 175
- System.Runtime.InteropServices.RuntimeMethodHandle.GetFunctionPointer method, 178
- System.Runtime.Remoting namespace, 274
- System.Runtime.Remoting.Context.IContextAttribute interface, 223
- System.Runtime.Remoting.Contexts.Context type, 235
- System.Runtime.Remoting.Contexts.IContextProperty interface, 239
- System.Runtime.Remoting.ILogicalThreadAffinative interface, 234
- System.Runtime.Remoting.Messaging.CallContext type, 232
- System.Runtime.Remoting.Messaging.IMessageSink interface, 228
- System.Runtime.Remoting.Messaging.ReturnMessage class, 211
- System.Runtime.Remoting.Proxies.RealProxy type, 214
- System.Runtime.Remoting.Services class, 214
- System.Runtime.RemotingServices.ObjRef object, 274
- System.RuntimeTypeHandle type, 79, 84
- System.SecurityException exception, 32
- System.Security.IPermission interface, 304–306
- System.Security.Permissions.CodeAccessSecurityAttribute, 315
- System.Security.PermissionSet method, 304
- System.Security.Permissions.IUnrestrictedPermission interface, 307–308
- System.Security.Permissions.SecurityAction, 315
- System.Security.Permissions.StrongNamePublicKeyBlob wrapper object, 287
- System.Security.Policy namespace, 285

- System.Security.Policy.Evidence type, 289, 314
- System.Security.Policy.IMembershipCondition interface, 295
- System.Security.Policy.PolicyLevelType enumeration, 291
- System.Security.Policy.PolicyLevel type, 293
- System.Security.Policy.UnionCodeGroup type, 296
- System.Security.SecurityException exception, 313
- System.Security.SecurityManager.PolicyHierarchy type, 293
- System.Security.SecurityZone enumeration, 284
- System.Security.SuppressUnmanagedCodeSecurity attribute, 346
- System.SerializableAttribute pseudo-custom attribute, 105
- System.Single type, 197
- System.String type, 18, 114, 128, 351, 353–354
- System.SystemException exception, 200
- System.Text.StringBuilder type, 354
- System.Threading.Interlocked type, 196
- System.Threading.Monitor type, 197–198
- System.Threading.Thread type, 254
- System.Threading.Thread.CurrentContext property, 235
- System.Threading.ThreadPool type, 371
- System.Threading.ThreadPool.GetMaxThreads method, 193
- System.Threading.WaitHandle type, 191, 197
- System.Type type, 82–83, 96, 110, 181
- System.Type.BaseType property, 86
- System.Type.GetConstructor method, 93
- System.Type.GetConstructors method, 93
- System.Type.GetEvent method, 99
- System.Type.GetEvents method, 99
- System.Type.GetInterfaces method, 85
- System.Type.GetMember method, 89, 91
- System.Type.GetMembers method, 87
- System.Type.GetMethod method, 94
- System.Type.GetMethods method, 94
- System.Type.GetProperties method, 97
- System.Type.GetProperty method, 97
- System.Type.GetType method, 96
- System.Type.IsAssignableFrom method, 85
- System.Type.IsSubclassOf method, 85
- System.Type.TypeInitializer property, 93
- System.UIntPtr type, 333
- System.ValueType base type, 114, 116, 117
- System.WeakReference type, 145
- System.Xml assembly, 23
- Target methods, address pushed on stack at runtime, 176
- Target object reference, 178
- Target property, 146
- Termination handlers, 148, 203, 383
- this keyword, 70, 71, 237
- Thread.CurrentContext, 245
- Threads, 254
 - binding call context to, 232–233
 - boosting priority, 247–250
 - contexts, 235
 - exclusive locking, 197
 - hard, 254–256
 - manipulating state of, 243
 - monitor-based locking, 197–198
 - not handling exceptions, 260–261
 - number in thread pool, 193–194
 - priorities, 221–222, 240–241
 - shared locking, 197
 - soft, 254–256
- throw instruction, 199–200
- Thunking layers, 327–328
- TLB (Type Library) files, 3, 4, 383
- tlbexp.exe, 358, 371, 383
- tlbimp.exe, 360, 383
- TLS (thread local storage), 255–256, 383
- Top-of-stack permission sets, 314
- ToString method, 113, 355
- TP (transparent proxy), 213–217
 - exposing public fields of classes, 220
 - method invocations, 217
- Transactional COM+*, 368
- Transition frames, 343–345
- Trusted components permissions, 319–320
- Trustworthy sites, 285
- try-finally statement, 203
- Turn method, 170
- Type compatibility tests, 84–85
- Type definitions, 1, 6
- Type handle, 79, 82, 165
- Type initializers, 60–62, 93, 383

- Type-centric, 9
- Type-compatible array types, 135
- TypeHandle property, 82
- typeof operator, 82–83
- TypeResolve event, 265
- Types, 49
 - abstract, 70
 - affiliation with instances, 77
 - affiliation with objects, 78
 - AppDomain-Bound, 275
 - AppDomains, 253
 - base types, 69–75
 - blittable, 350–351
 - categories, 64–69
 - compatibility tests, 84–85
 - complete description of, 79
 - constructors, 62–63, 93
 - contract, 51
 - declaring compatibility with interfaces, 66
 - default constructor, 62
 - derived, 70–71
 - distinguished methods, 60
 - dynamic coercion, 79–80
 - enumerating interfaces, 85–86
 - exact memory layout, 53
 - explicit type initializer method, 61, 62
 - fields, 51–52, 91–92
 - formatted, 335
 - fundamentals, 49–60
 - initialization, 60–63
 - initialization error at runtime, 335
 - initializing, 31–38
 - instances, 77–86
 - interface offset table, 166
 - interface table, 80
 - interfaces, 64–69
 - internal, 21, 22
 - IPatient category, 65
 - layout of, 334–337
 - loading, 31–38
 - method table, 158
 - methods, 54–56, 157–165
 - multiple method definitions, 58
 - nonblittable, 350–351
 - non-interface, 70
 - non-opaque, 338
 - object serialization, 275
 - on-demand loading, 31
 - opaque, 337–339
 - ordered instances, 128–129
 - permissions, 306–307
 - private, 21
 - properties, 96
 - protected, 21–22
 - proxiability, 219–227
 - public, 21
 - reading definitions, 86–96
 - remote-unaware, 275
 - runtime, 77–86
 - sealed, 70, 74
 - static field declarations, 61
 - translating definitions, 357–360
 - type initializers, 60–61, 93
 - unable to resolve, 264–265
 - unbound, 275
 - uninitialized instance, 224
 - universal, 65
- UIPermission permission, 322
- Unbound types, 275
- Unboxing, 133
- UnhandledException event, 260
- UnhandledException event handler, 261
- Union method, 307
- Unique identifier, 277–278
- Universal types, 65
- Unmanaged execution, 341
- Unmanaged methods, 341–343
- Unmanaged modules, 345–361
- Unmanaged pointers, 329–331, 332
- UnManagedCode permission, 346
- UnmanagedType type, 351
- Unmarshal method, 274
- Unrestricted permissions, 307–308
 - /unsafe+ command-line switch, 330–331
 - /unsafe command-line switch, 330–331
- Unwrap method, 278, 279
- Up-casting, 79
- Upper bound, 193
- URI (Uniform Resource Identifier), 277–278, 383
- URL (Uniform Resource Locator), 31, 285–286, 383
- Url type, 283, 284
- URN (uniform resource name), 383
- urn:schemas-microsoft-com:asm.v1
 - namespace, 34
- URT (Universal Runtime), 6
- User policy level, 291, 292

- User-defined types, 77
- User-defined value types, 114
- using statement, 151–152
- utilities.dll components, 25

- Valid object, 119
- ValidOn property, 111
- __value modifier, 338
- Value type variables, 119–120
- Value types, 113
 - arrays, 137
 - assigned to object reference variable, 133
 - assignment, 121–122
 - class fields, 119
 - default implementation of
 - Object.Equals, 127
 - defining, 116
 - enumeration types, 117
 - fields, 114
 - instances, 119
 - local variable, 119
 - methods, 114
 - parameters, 122–123
 - restrictions, 114–115
 - System.ValueType base type, 114
 - user-defined, 114
- Values, 113–119
- Variables, 102, 119
- VB.NET, 12
 - build-in types, 114
 - ByVal or ByRef parameter modifier, 56
 - custom attributes, 105
 - defining delegate types, 179–180
 - delimiter characters, 60
 - Function keyword, 54
 - hide-by-name types, 72
 - hide-by-signature types, 72
 - indexed properties, 102
 - jagged arrays, 140
 - /main command-line switch, 17
 - Main method, 16
 - New subroutine, 62
 - Shadows keyword, 71
 - Shared keyword, 51
 - Sub keyword, 54
 - /target command-line switch, 14–15
 - this or Me variable, 58
 - type initializers, 60–61
 - WithEvents modifier, 100
- vcb.exe, 14, 383
- Verifiable code, 282–283, 330
- Version policies, 33, 384
 - configuration files, 34, 36–37
 - partially-specified version names, 34
 - per application, 36–37
 - per component, 36–37
 - per machine, 36–37
 - turning off, 38
- Version property, 286
- Versioned types, 47
- Versioning hazards, 44–47
- virtual metadata attribute, 159
- Virtual methods, 165–172, 224
 - mandating or prohibiting replacement, 162
 - method table, 159
 - method table offsets, 160
 - news slot attribute, 160
 - overriding, 164–165
 - replacement for virtual method of base type, 162–164
 - slots, 160, 162
- Virtualization, xvii–xix
- Visual Basic, 102
- Visual C++
 - delay-load feature, 31, 156
- Visual Studio.NET, 384

- Weak references, 145–146
- WeakReference.Target property, 145
- web.config files, 34
- WebPermission permission, 32, 302
- wincv.exe, 384
- Worker processes, 273
- Worker threads, 193–195
- wSDL.exe, 384

- X.509 certificates, 287–289
- XML (Extensible Markup Language), 34
- XML Web Services, xviii
- xsd.exe, 384

- Zone type, 283, 284–285