Index

NOTE: Page references followed by *f* and *t* indicate figures and tables, respectively.

A

Abstract classes, 34–36
  code implementations and, 59
  defining, 34–35
  vs. interfaces, 59–60
  in Remoting, 222
  use of, 34
  virtual, 107
Abstract factory pattern, 555
Abstract members, 35
Abstract methods, 60
`AcceptChanges,` in `DataSet` update, 371
Access Control List (ACL), users added to, 449–454
Access member, invocation of, 9
Access modifiers
  default, 26
  definition of, 24
  in information hiding, 27, 43–45
  in interfaces, 49
  rules for, 26, 44–45
  using, 24–26
ACL (Access Control List), users added to, 449–454
`Activator` class, instance created with, 94–95
ActiveX controls
  importing, 214, 215*f*, 216*f*
  smart-updating of, 349–350
Adapter
  `DataSet` use of, 360–362
  for parameterized values, 368
  in stored procedures, 388
Adapter pattern, 46
`AddHandler` statement, 74–75
`Add` method, in returning typed collection, 485, 486*f*
`AddNew` method, 381, 386
ADO.NET
  about, 351–352
  command builder in, for SQL generation, 367–368
  command objects in, 366–367
  database connections in, 353–360
  database relationships in, 364–366
  `DataReader` class in, 362–363
  `DataSet` in, 360–362
    adding data to, 371–373
    filtering and sorting, 373–376
    updating, 369–371
  `DataTable` in, 363–364
  `DataView` in, 363–364
    updating, 380–387
  fundamentals of, 352–353
  inheritance from, 385
  interfaces in, 554–555
    programming with, 421–422
  provider classes for, 352–353
  stored procedures in, programming with, 387–403
  transactions in, 407–410
Aggregation
  definition of, 28
  vs. inheritance, 45–48
  relationship in, 45
Algorithm, interfaces and, 577
`AllowDelete` property, 380
`AllowEdit` property, 380
`AllowNew` property, 380
`AllowPaging` property, 572
`AllowSorting` property, 573
Application caching, *vs.* session caching, 531
Application configuration file
  Boolean switch defined in, 626–627
  for server application, 232–233
Application domain, Remoting and, 229
Application domain security policy, 648
Application-level event handlers, 516–518
Application loader
  generic, 336–337
  for thin clients, 323–324
Application log, definition of, 628
Application state, use of, 519
`appSettings,` connection string externalized with, 385, 552
Array
  binding to `DataGrid,` 574–577
  sorting, `IComparer` for, 577–579
  typed collection as, 481–482
`ArrayList,` in binding, 110, 112
`.asmx` file
  copying, 444
  for Web Service, 441, 462, 462*f*
ASP, migration from, 685–686
ASP.NET
  caching in, 518–540
  cascading style sheets in, 494–497
  debugging in, 635–636
    reenabling, 357, 358*f*
  event handlers in, application level, 516–518
  user controls in, 503–516
  Web service application in, 436, 436*f*
ASP.NET applications, screen layout for, designing, 492–503

- Assembly(ies)
  - consuming, in VB6, 205–207, 206*f*
  - conversion of, to type library, 203–205
  - in copy-and-paste deployment, 323–324
  - definitions of, 132
  - downloading, 470
    - permissions for, 653–655, 653*f*, 655*f*
  - fat objects and, 477–478
  - IL code in, 125–126
  - metadata in, 126
  - for policy, 652
  - registration of, 203–204
  - in Remoting, 225–229, 225*f*
  - sandboxing, for testing, 664–665, 665*f*
  - security asserts and, 663
  - in thin client, 326–327
    - downloading, 334
    - security and, 333–334, 333*f*
  - unregistration of, 205
- Assembly attributes
  - for application information, 132–141, 136*f*
  - reading, Reflection for, 132–136
  - using, 132–133
- AssemblyAttributeVersion, Reflecting, 137
- Assembly class
  - loading, at runtime, 91–93
  - in thin client, 326
- Assembly keyword, 132
- Assert method
  - behavior of, 665–669
  - in Code Access Security, 663–669, 671
  - for debugging, 615, 617–618, 618*f*
- Association, definition of, 28
- Asynchronous behavior
  - completing, 164–167
  - execution of, 161
  - for file operations, 169–170
  - in .NET framework, 167–171
  - programming, 163–164
  - *vs.* Timer control, 170–171
  - Web method and, 488
  - for XML Web Services, 167–169, 433, 486–489
- Asynchronous Remoting, 261
- @ blocks, in cascading style sheets, 497
- @ prefix, 393
- Attribute(s). *See also* DefaultMemberAttribute
  - applying, 127–132
  - for bitmap association, 274
  - classes and, 126
  - of controls, modifying programmatically, 502–503
  - conventions for, 126–127
  - custom
    - creating, 141–143
    - Reflection of, 113–114
  - emitting
    - with CodeDom classes, 153–157
    - to IL, 150–153, 150*f*
  - of HTML tables, 493
  - notation for, 127
  - parameters for, 127
  - Reflection of, 144–150
    - custom, 113–114
  - security and, 157
  - usage of, specifying, 141–143
  - in VB6, 66
- Attributes collection, 503
- Attribute suffix, 127
- AttributeTargets, 132
- Authentication
  - cookies for, 548
  - for Web applications, 546–548
  - in Web.config, 446, 448, 547
- Authorization, in Web.config, 454–456
- Automatic navigation, Reflection for, 112
- Automation Model, for debugging, 641
- AutoResetEvent, for blocking, 173, 178
- Autos window, for debugging, 605
- Available Mbytes counter, 633
- aximp.exe, 214, 215*f*, 216*f*

B

- Base class. *See* parent class
- Base class libraries (BCL), 574
- BeginInvoke method
  - for asynchronous behavior, 163
  - for asynchronous Remoting, 261
  - for file read, 170
  - IAsyncResult and, 165
  - Windows Form controls and, 180
- Begin method
  - for asynchronous behavior, 170
  - in Web methods, 486–487
- Bézier curves, 39
- Binary assemblies, downloading, 470
- Binder class
  - about, 99
  - ArrayList and, 110, 112
  - custom, 107–109
  - inheritance in, 99, 107–109
- Binding. *See* data binding
- BindingFlags enumerator, in Reflection, 101–102
  - of nonpublic fields, 106
  - properties and, 104
  - of public fields, 105
- Bitmap, for custom control, 273–276, 275*f*, 276*f*
- Black-box testing, of Web Services, 439–440
- Blocking
  - in assembly loading, 92–93
  - AutoResetEvent for, 173, 178
  - with EndInvoke, 165–166
  - ManualResetEvent for, 173, 178
  - with resource protection block, 363, 402
  - with WaitHandle objects, 166–167
- Boolean switches, debug code managed with, 625–628
- Bound columns. *See* columns, binding
- Boxing, 6, 10–13, 12*f*
- Brackets, for keywords, 92
- Breakpoints
  - conditional, 612, 613*f*
  - managing, 611–613, 612*f*, 613*f*
  - in Web applications, 637
- BrowsableAttribute, 284, 287–288
- Build-and-browse menu, for Web Services testing, 440–441
- Builder classes, 115–116, 121
  - for emitting IL, 152
- By-reference objects
  - *vs.* by-value objects, 248
  - event handler attached to, 254
  - for Remoting, 223–239
- Byte code, in Java language, 29
- By-value classes, as by-value objects, 239–242

- By-value objects
  - *vs.* by-reference objects, 248
  - for Remoting, 239–248
  - revising for, 242–243

C

- `cache` class, 537
- Caching. *See also* session caching
  - `HttpApplicationState` class in, 519–530
  - importance of, 518
  - of objects, 518–540
  - session *vs.* application, 531
- `Calculate` method, synchronous call to, 161–163
- Calculator, code for, 594–604
- Callback function, 73–74
  - delegates and, 74, 82
- Call stack, security asserts for, 663
- Call Stack window, for debugging, 608, 609*f*
- `CancelEdit` method, 381, 386
- `CanConvertFrom,` 304–305
- `CanConvertTo,` 304–306
- `CanExtend,` 313
- CAO. *See* Client-Activated Objects
- Cascading style sheets, for ASP.NET pages, 494–497
- Case-sensitivity, in Visual Basic .NET, 14
- `caspol.exe,` 338, 649–650, 650*f*
- `Catch` block, 92–93
- CCW. *See* COM callable wrapper
- Channel
  - for client, 231, 255
  - for server, 230, 256
- Character encoding, 170
- Child class
  - definition of, 28
  - inheritance of, 33
  - reason for, 33
  - synonyms for, 32
- Class(es), 8–10. *See also* Metaclasses
  - abstract (*See* abstract classes)
  - access modifiers for, 26
  - attributes and, 126
  - Code Access Security for, 647
  - control and, 19
  - definition of, 5–6, 8, 27
  - event declared in, 80–81
  - event implementation in, 23
  - exposing to COM, interfaces for, 211
  - in imported COM code, 194–195
  - inheritance of, 33–34
  - instantiating, 9
  - nested, 79, 417
  - *vs.* objects, 9
  - proxy (*See* proxy class)
  - public members in, 28, 43
  - support in, 8, 13
  - threading and, 174–175
- Class(es), in cascading style sheets, 497
- Class header, in inheritance statement, 33
- `Class_Initialize` method, *vs.* constructors, 17
- `ClassInterfaceAttribute,` 211
- Class library
  - COM, debugging, 216–217, 217*f*
  - creating, 202–203
  - debugging, 217–218
  - Toolbox and, 271
- Client, remote
  - channel for, 231, 255
  - event handlers on, 249
  - requests from, 229–232
  - revising, for by-value objects, 242–243
  - server and, relationship between, 232*f*
- Client-Activated Objects (CAO)
  - implementation of, 237–239
  - for Remoting, 234–239
  - Server-Activated Objects *vs.,* 235
  - server for, 235–236
- Client application, 224–225, 225*f*
  - configuration file for, 255–256
  - event registration by, 256
  - implementation of, 237–239, 256–259
  - migration of, from VisualBasic 6, 677–678, 678*f*–685*f*
  - Web shared directory for, 325
- CLR. *See* Common Language Runtime
- CLS (Common Language Specification), 5
- Code, managed *vs.* unmanaged, 646
- Code Access Security
  - about, 645–646
  - attributes and, 157
  - demands in, 660–663
  - Windows security and, 645–646
- CodeDOM
  - attributes emitted with, 153–157
  - dynamically loaded controls and, 510–511
  - source code from, 413
  - WSDL and, 469
- Code Editor, event handlers in, 67
- Code generators, 153–156
- Code group
  - managing programmatically, 341–344
  - in runtime security policy, 652
  - security permissions for, 338–341, 339*f*, 340*f*
- Code implementations
  - abstract classes and, 59
  - interfaces and, 49, 59
- Code testing. *See* testing
- Collection
  - indexer in, 479, 481–482
  - strongly typed
    - about, 479–482
    - implementation of, 479–480, 519–522
    - navigation links as, 508–510
    - returning, 478–486
    - serialization of, 482–485
    - storing, in application cache, 523–524
  - typed, 479
    - as array, 481–482
- `ColorInfo` class, 516–518
- Columns
  - binding, to `DataGrid,` 569–571, 570*f*, 571*f*
  - template, 586–588
- COM (Common Object Model)
  - calling from .NET, 188–202
  - calling .NET from, 202–213
  - early bound, 194–196
  - exposing delegates to, 207–210
  - exposing types to, 203–205
    - debugging and, 217–218
  - imported, 191–196, 194*f*
  - late bound, 197–198
  - parameterized constructors and, 202
  - support for (*See* COM Interop)
  - test object in, 189–191
  - viewing, 191, 192*f*
- COM callable wrapper (CCW), 202
  - exporting to, 211–213, 212*f*

COM class libraries, debugging, 216–217, 217*f*
COM Class template, 211–213, 212*f*
COM components, smart-updating of, 349–350
COM events
  consuming, 198–202, 200*f*
  delegates for, 198–200, 200*f*
    multicast, 200–202
COM Interop
  attributes for, 210–211
  for calling .NET code, 202
  debugging, 214–218
  error handling in, 213–214
  migration with, 674
Command builder
  for `DataSet` updating, 369–371
  SQL generated with, 367–368
Command objects
  `DataSet` filled by, 366–367
  in stored procedures, 387–388
Command window, for debugging, 609–611, 610*f*
Common Language Runtime (CLR)
  inheritance of, 31
  managed code and, 646
Common Language Specification (CLS), 5
Common Object Model. *See* COM
Compiled regular expressions, 117–118
  emitting dynamically, 122–123
Compiler, *vs.* IL, 126
Complex type
  flattening, 468
  implementation of, 463–466
`ComponentDesigner` class, 317
Components, *vs.* controls, 264
Components, custom
  implementation of, 264–267
  inheritance in, 264
  testing, 270–271, 271*f*
Composite user control
  data binding and, 581
  defining, 579–582
  definition of, 503
  implementation of, 580–581, 580*f*
  repeating, `DataList` for, 579–585
  Web page with, 582–585, 583*f*
Composition. *See* aggregation
`ComRegisterFunctionAttribute,` 210
`ComUnregisterFunctionAttribut`, 210
`ComVisibilityAttribute,` 210–211
`ConditionalAttribute,` for connection testing, 556–557
Conditional breakpoints, 612, 613*f*
Configuration file
  for client application, 255–256
  for connection string, 356–357
  for security policy, modifying, 649
  for server application, 232–233
  for server registration, 233–234
  for Web Services, default, 445–446, 446–448
  for Windows applications, Boolean switch defined in, 626–627
Configuration utility, for security management, 338–341, 339*f*, 340*f*, 651
Connection
  to database, 551–557
  `DataSet` use of, 360, 371
  security and, 671
Connection pooling, in .NET, 359
Connection string
  centralizing, 554
  contextual wrapper for, 358–359
  database providers in, 354, 379
  defining, 354–356, 355*f*
  externalizing, 552
    with `appSettings,` 385
  reading, 357, 553
  for SQL server, defining, 390–391, 391*f*
  storing, 356–357
  testing, 358–360, 555–556
  in `Web.config` file, 552–553
Constructor(s)
  adding, 17–19
  arguments passed to, 98
  definition of, 17
  for deserialization, 244, 247–248
  dynamically invoking, 95–98
  inherited, 42–43
  parameterized, 18–19
    COM and, 202
    for Remoting, 237
  `UserControl` object and, 48
Consumer, of Web Service, implementation of, 469–471
Contract, for interfaces, 49
Control. *See also* control, custom; user control; Windows control
  attributes of, modifying programmatically, 502–503
  binding, to methods, 527
  vs. components, 264
  composite (*See* composite user control)
  position of, HTML tables for, 492–494
  properties bound to, Reflection in, 110, 111*f*
  source code for, 270
  style applied to, 501–502
Control, custom
  adding to Toolbox, 271–273, 272*f*, 273*f*, 274*f*
  attributes of, 283–288
  bitmap for, 273–276, 275*f*, 276*f*
  documentation for, 277–278, 277*f*
  implementation of, 267–271
  properties of
    categorizing, 293–294, 294*f*
    default value for, 293
    editing, 288
    type conversion for, 294–310
  testing, 270–271, 271*f*
  type editor for, 288–293
  for Web, 320–321
`ControlDesigner` class, 314–317
`ControlHintAttribute,` 143, 144, 148–149
`ConvertFrom,` 304–305
`ConvertTo,` 304–306
Cookies
  for authentication, 548
  session ID and, 532, 534, 534*f*
  use of, 537–538
Copy-and-paste deployment, 323–324
Copy Project, for Web Service deployment, 442, 442*f*
`CreateInstance,` 326
Custom attributes
  creating, 141–143
  Reflection of, 113–114
Custom binder, implementation of, 107–109
Custom components. *See* components, custom
Custom control. *See* control, custom

- Custom event log
  - for debugging, 264–267, 628–631
  - information sent to, 631, 632*f*
- Custom log, definition of, 628
- Custom user control. *See* user control, custom

D

- Database
  - connections to, 551–557
    - in ADO.NET, 353–360
    - simultaneous, 351
  - relational
    - normalization of, 407
    - object-oriented systems and, 478
- `Database` class, interfaces for defining, 553–555
- Database providers
  - for ADO.NET, 352–353
  - in connection string, 354, 379
  - interfaces and, 422, 555
- Database relationships, defining, 364–366
- Data binding
  - composite user control and, 581
  - to custom user control, 281–283
  - early, for COM objects, 194–196
  - in HTML editor, 568–569
  - late
    - for COM objects, 197–198
    - definition of, 88
    - disadvantages of, 198
    - `Option Strict On` and, 89–90
    - in Reflection, 87–90
  - for multi-value Web controls, 569–571
  - serialization for, 575
  - for single-value Web controls, 562–569, 563*f*
- `DataGrid`
  - for adding data, 371
  - binding array object to, 574–577
  - binding data to, 569
  - bound columns of, 569–571, 570*f*, 571*f*
  - changes to, 371
  - CRUD operations performed by, 386
  - dynamically loaded, 513–514, 513*f*
  - for paging, 572–573, 574*f*, 587, 587*f*
  - for sorting, 573–579, 574*f*
- Data Link properties, for connection string, 355, 355*f*
- `DataList` control
  - binding data to, 569, 581–582
  - for navigation links, 505
  - for repeating composite controls, 579–585
  - for repeating user controls, 527–530
  - template for, 528, 528*f*
  - Web page with, 582–585, 583*f*
- `DataReader` class, use of, 362–363
- `DataRelation` object
  - in `DataSet,` 380
  - for relationship defining, 363, 364–366
  - version of, 360
- `DataRow` object
  - in filtering, 376
  - in sorting, 376
- `DataSet` class (object???)
  - adding data to, 371–373
  - as base class for object, 478
  - command objects and, 366–367
  - connection of, 360, 371
  - database providers and, 353
  - `DataTable` and, 363–364
  - filtering, 373–376
  - sorting, 373–376
- `DataSet` object (class???)
  - fat, 477
  - filling, with adapter, 360–362
  - inheritance from, 385
  - returning, 470–477
  - serialization into, 471–474
  - serialization of, 419–421
  - storing, in session cache, 535–536
  - typed, 372
    - creating, 410–419
    - generating, 414–417, 416*f*
    - initializing, 417–419
    - multiple tables added to, 412–413
    - programming with, 417–419
  - updating, 369–371
  - version of, 360
- Data source, in connection string, 354
- `DataSource` property, in data binding, 569
- `DataTable` object
  - in `DataSet,` 380
  - default data view in, 380
  - definition of, 361
  - filtering with, 376
  - inheritance from, 385
  - row added to, 371–373
  - sorting with, 376
  - use of, 363–364
  - version of, 360
- Data view
  - custom, 558–561, 561*f*
  - default
    - binding to data grid, 557–558, 558*f*
    - in `DataTable,` 380
- `DataView` class, 380, 557–561
- `DataView` control, binding data to, 562–568, 567*f*
- `DataView` object
  - filtering, 373–376
  - sorting, 373–376
  - updating, 380–387
  - use of, 363–364
  - version of, 360
- `DateTime` variable, 7
- Davis, Phillip, 311
- `DbType` property, 393–394
- DCOM, configuring, for stored procedure debugging, 405, 406
- Deadlock, in multithreading, 177
- `Debug` class, 614–615
- Debugger
  - integrated, 438–439, 440*f*, 441*f*
  - non-integrated, 642
- Debugging
  - application event log for, 628–631
  - in ASP.NET, reenabling, 357, 358*f*
  - `Assert` method for, 615, 617–618, 618*f*
  - Automation Model for, 641
  - Autos window for, 605
  - Boolean switches for, 625–628
  - Call Stack window for, 608, 609*f*
  - in COM Interop, 214–218
  - Command window for, 609–611, 610*f*
  - Disassembly window for, 608, 609
  - event log for, 264
  - guidelines for, 403

- Debugging, *continued*
  - Immediate window for, 609–611
  - Locals window for, 605
  - Memory window for, 608
  - of multi-language applications, 638–641
  - Output window for, 611
  - performance counters for, 631–634
  - practice code for, 594–604
  - `Process` class for, 634–635
  - QuickWatch window for, 606, 607*f*
  - of running processes, 635–636
  - of stored procedures, 403–407
  - `Trace` class for, 618–619
  - Watch window for, 605, 606–607, 608*f*
  - of Web application, 636–638
  - of Windows application, 636–637
  - of XML Web Services, 438–441
- Debugging Object Model, 641
- `DEBUG` token, for connection testing, 556, 556*f*
- Declaration, in style block, 498
- Declarative caching, 539
- Declarative security
  - advantages of, 659
  - attributes and, 157
  - definition of, 656
  - exceptions in, 658, 658*f*
  - *vs.* imperative security, 655–660
- `DefaultBinder` property, 99
- Default data view, in `DataTable`, 380
- `DefaultMemberAttribute`, 99–100, 128–129
- `Default` modifier, 129–130
- Default properties, 319–320
- `DefaultPropertyAttribute`, 129, 320
- `DefaultTraceListener`, 619
- Default value
  - for control properties, 293
  - in Reflection, 292
- `DefaultValueAttribute`, 292
- Dehydrated classes, 477
- Delegate(s)
  - for asynchronous Remoting, 261
  - callback function and, 74, 82
  - for COM events, 198–200, 200*f*
    - multicast, 200–202
  - consuming, in COM, 209–210
  - conversion of, to COM connection points, 207–209
  - creating, 83
  - defining, 83–84
  - definition of, 63, 82
  - events as, 80–81
  - existing, 84–85
  - exposing to COM, 207–210
  - *vs.* method heading, 83
  - multicast, 200–202
  - for multithreading, 171–172, 178
  - procedure signature and, 84
  - Remote events and, 249
  - use of, 82–83
- `Delegate` class, 22–23
- `Delete` method, 386
- `Demand` method, 660–663
- `Deny` action, 669
- Derived class. *See* child class
- `DescriptionAttribute`, for custom control, 277, 277*f*
- Deserialization
  - automatic, 242–243
  - constructor for, 244
  - `GetObjectData` and, 485
- Designer, for Windows control, 314–319, 317*f*, 318*f*
- `DesignerAttribute`, 317
- `DesignerSerializationVisibilityAttribute`, 283–284, 285–287
- Design patterns
  - abstract factory, 555
  - definition of, 524
  - interfaces as, 422
  - Iterator pattern, 50
  - purpose of, 46
- Design time
  - event handler association at, 73
  - type conversions at, 307–310
- Deterministic memory management, 20
- `Diagnostics` namespace, 614–615
- `dim` statement, for variables, 3–4
- Disassembly window, for debugging, 608, 609
- Discovery (.DISCO) files
  - definition of, 428
  - in Web Service deployment, 444–445
- `Dispose` method
  - implementation of, 21–22
  - overloading, 22
- DLL file, for Web Service, 167, 441, 444
- Document Outline, for cascading style sheets, 496–498, 497*f*
- Document Styles, for linking style sheets, 501, 502*f*
- Dot operator (`.`)
  - access member invocation with, 9
  - Intellisense and, 4
  - operation invocation with, 9
- Dynamic filtering, 373–376
- Dynamic sort behavior, 373–376
  - callback function for, 76–80

E

- Early binding, for COM objects, 194–196
- Edit and continue behavior, 614
- `EditorAttribute`, 292
- `EditorBrowsableAttribute`, 283–285, 285*f*, 286*f*
- `EditValue`, 291–292
- Elements, in cascading style sheets, 497
- Emitting code
  - for attributes, 150–153, 150*f*
  - benefits of, 118
  - at runtime, 114–123, 115*f*
- Encapsulation
  - definition of, 27
  - `MyBase` keyword and, 37
- `EndEdit` method, 381, 386
- `EndInvoke` method
  - for asynchronous Remoting, 261
  - blocking with, 165–166
  - Windows Form controls and, 180
- `End` method
  - for asynchronous behavior, 170
  - in Web methods, 486–487
- Enterprise security policy, 649
- Enumeration, abstract classes and, 35
- `Enum` method, code for, 16
- Equality testing, of structures, 8
- Error handling, in COM Interop, 213–214
- Event(s)
  - access modifiers for, 26
  - adding, 22–24
  - as delegate, 80–81
  - handling, 23–24 (*See also* event handler)
  - in interfaces, adding, 55–56

  multiple, event handler for, 68–69, 71–72
  multiple handlers for, 72
  Reflection of, 109–110
  remote
    behavior of, 250
    handling, 249–259
    invoking, 250–259
    registration of, 256
  return types and, 80
  in Windows operating system, 63
`EventArgs` argument, 70–71, 84
Event handler
  `AddHandler` statement for, 74–75
  application level, 516–518
  association of
    at design time, 73
    at runtime, 73, 75–80
  for by-reference objects, 254
  on client, remote, 249
  in Code Editor, 67, 68*f*
  coding, habits for, 71–72
  for extender provider, 313–314
  in Form Designer, 64–67
  generating, 24, 25*f*
  method inheritance and, 67, 68*f*
  multiple, 72
  for multiple events, 68–69, 71–72
  for paging, 572–573
  for Remoting, 258
  signature for, 67, 74
  for sorting, 573–574
  thread for, 259
  in VB6, 66
`EventHandler` class, 69
  as delegate, 81, 84
Event log
  custom
    for debugging, 264–267, 628–631
    information sent to, 631, 632*f*
  for debugging, 264
  information sent to, 629–630
  for tracing, 264, 622, 623*f*
  writing to, 249
`EventLog` class, versions of, 9
`EventLogTraceListener`, 619
Event signatures
  defining, 80
  event handlers and, 69
  in interfaces, 50
  types of, 80
Exceptions
  COM and, 213–214
  in declarative security, 658, 658*f*
  file not found, 92–93
  in imperative security, 658, 658*f*
  in Remoting, 259
  in type conversion, 301
Expressions, definition of, 607
eXtended Markup Language. *See* XML
Extender providers, implementation of, 310–314
External style sheet
  for ASP.NET pages, 496–498, 497*f*
  in cascading, 501
  linking to, 501, 502*f*
  style block in, 498–499, 498*f*, 499*f*
  visually building, 499–500, 500*f*

F
Façade, for security, 647
Factoring, 49
Factory methods, 522
Fat objects
  `DataSet` as, 477
  definition of, 468
  proxy class and, 477–478
Fields
  access modifiers for, 26
  access to, 14
  adding, 13–14
  definition of, 13
  Reflection of, 104–109
`FileNotFoundException`, 92–93
File operations
  asynchronous invocation of, 169–171
  synchronous invocation of, 170
File read, asynchronous, 169–170
`FileStream` trace listener, 620–622
`FileVersionInfo` class, 132, 137–139
Filtering
  of `DataSet`, 373–376
  in XSLT, 542
`Finalize` method
  implementation of, 20–21
  properties of, 21
Flattening, of complex types, 468
Football game, 331–333, 332*f*
Form. *See* Windows Forms
Form Designer
  for ASP.NET, 492
  event handlers in, 64–67
Friend access, 26, 44
Function pointers, *vs.* delegates, 82

G
Garbage collection
  definition of, 20
  imported COM code and, 197
Generalizer
  definition of, 25
  inheritance used by, 270
`Get` blocks, read-only and, 15
`GetCustomAttributes` method, 113
Get methods
  for attributes, 113
  extender providers and, 313
  plural forms of, 97
  for property Reflection, 103–104
  for serialization, 244, 247–248, 287, 485
  for services, 291
  for type information, 91, 301–302
`GetObjectData`, for serialization, 244, 247–248, 485
`GetService` method, 291
`GetType`, for type information, 91
`GetTypeCode`, 301–302
Grandparent class, 32
Grid Layout panel, for control position management, 492
Groups, in Windows authentication, creating, 451–454, 451*f*, 452*f*, 453*f*

H
Hackers, 643
`Handles` clause, 68–69
Hejlsberg, Anders, 84, 222
Hollis, Billy, 323
`HResult`, exceptions and, 213–214
HTML (HyperText Markup Language), debugging, 637
HTML controls, as server controls, 511–512, 512*f*
HTML editor, data binding in, 568–569
HTML Flow Layout, for control position management, 492

- HTML tables
  - context menu for, 494*f*
  - for control positions management, 492–494, 492*f*, 504–505
- HtmlTextWriter object, 321
- HttpApplicationState class, 519–530
- HttpChannel constructor, 230
- HttpContext object, 516–518, 537
- HTTPS, thin client applications and, 350
- HttpSessionState class, 531–537
- HyperText Markup Language. *See* HTML

I

- IAsyncResult interface, 165, 166
- ICollection interface, 485
- IComparer interface
  - for array sorting, 577–579
  - for dynamic sort behavior, 78–80
- IConvertible interface, 295–302
  - behavior of, 306–307
  - conditions for, 295
- IDbCommand interface, 422
- IDesigner interface, 314, 319
- IDisposable interface, 22
- IEnumerable interface, 50
  - binding and, 575
  - serialization and, 482, 483
- IEnumerator interface, 50
- IExtenderProvider interface, 310–314
- If conditional statements, *vs.* sentinels, 91
- IFormatProviders interface, 307
- IIS. *See* Internet Information Services
- IL. *See* Intermediate Language
- ildasm.exe, 12*f*, 29
- Immediate window, for debugging, 609–611
- Imperative caching, 539
- Imperative security
  - advantages of, 659–660
  - *vs.* declarative security, 655–660
  - definition of, 656
  - exceptions in, 658, 658*f*
- Implementations, in imported COM code, 194–195
- Implements statement, for IDisposable interface, 22
- Imports statement, COM importing and, 196
- Indexer
  - in collections, 479, 481–482
  - default properties and, 319
- Information hiding
  - access modifiers in, 43–45
  - definition of, 27
  - ease of use and, 43
- Inheritance
  - in Adapter pattern, 46
  - from ADO.NET, 385
  - *vs.* aggregation, 45–48
  - of binder class, 99, 107–109
  - of classes, 33
  - conditional invocation of, 269
  - of constructors, 42–43
  - in custom components, 264
  - definition of, 28
  - of interfaces, 56–58
    - multiple, 58–59
  - of methods, event handlers and, 67, 68*f*
  - relationship in, 45
  - single, 31
  - terminology for, 32
  - of TypeConverter class, 304
  - in typed DataSets, 410, 417
  - in user controls, 516
  - of Windows control, 267
- InheritanceDemand action, 669
- Inheritance polymorphism, definition of, 421
- Inherits keyword, 32–33
  - in interface inheritance, 56
- InitializeLifetimeService, 254, 259
- Initialize method, for constructor implementation, 18
- Initializers, for variables, 4
- Inline style, in cascading, 501
- In-process state servers, configuration of, 532–534
- Input parameters, for stored procedures, 391–394
- Installer file, security management with, 337–349, 650, 652–653
- Installer library, implementation of, 344–346
- Instantiation, lazy, 110
- Integer type, System.Int32 and, 13
- Integer variable, 7
- Integrated debugger, for Web Services testing, 438–439, 440*f*, 441*f*
- Intelligent Background Transfer Service, 350
- Intellisense, 4–5
  - EditorBrowsableAttribute and, 283
  - styles in, 496–497, 496*f*
  - Web Services and, 432
- Interface(s)
  - *vs.* abstract classes, 59–60
  - access modifiers in, 49
  - in ADO.NET, 554–555
    - programming with, 421–422
  - aggregation and, 46
  - for classes, exposed to COM, 211
  - code implementation in, 49, 59
  - Database class defined by, 553–555
  - defining, 49, 50–51
  - dynamic, with XML, 540–546
  - events in
    - adding, 55–56
    - declaring, 81
  - flexibility of, 422
  - implementation of, 49, 51–56
  - in imported COM code, 194–195
  - inheritance of, 56–59
  - members of, 50
    - access modifiers for, 26
  - multiple, inheritance of, 58–59
  - prefix for, 49
  - properties in, 50
    - adding, 54–56
  - in Remoting, 225–229, 225*f*
  - securing code with, 647
  - security attributes and, 49
  - for sort comparisons, 78–80
  - in stored procedures, 388
  - use of, 34
- Interface polymorphism, 421
- Interface programming, 137–141
- Intermediate Language (IL), 29
  - in assemblies, 125–126
  - *vs.* compiler, 126
  - emitting at runtime, 114–123, 115*f*
    - for attributes, 150–153, 150*f*
- Internet connection
  - simultaneous, 351
  - for thin clients, 329–330, 337

Internet Information Services (IIS), Remoting and, 230–231
Interop assembly, 188–189, 193, 193*f*. *See also* COM Interop
Interop attributes, 210–211
`InvokeDefaultMember`, 100
`InvokeMember` method
  `DefaultMemberAttribute` and, 128
  in Reflection, 101–102, 104
`Invoke` method
  for marshaling, 488–489
  synchronous call to, 162–163
  Windows Form controls and, 180
`InvokeRequired`, 160
`IPrincipal` interface, 658
`ISerializable` interface, 239
  implementation of, 243–248, 483–485
  `SerializableAttribute` and, 247
`IsPostBack` property, 588
`IsValidContext`, 291
`IsValidProvider`, 291
`Item` properties, 319–320
ItemTemplate, adding controls to, 582, 584*f*
Iterator pattern, 50

J
Java language, byte code in, 29
JITters, 29

K
Kernighan, Brian, 324
Keyboard hooks, 643
Keywords, brackets for, 92
Knopka, Ray, 273

L
Late binding
  for COM objects, 197–198
  definition of, 88
  disadvantages of, 198
  `Option Strict On` and, 89–90
  in Reflection, 87–90
Lazy instantiation, 110
Lifetime, for remote objects, 259
`LinkDemand` action, 669
Links. *See* navigation links
`ListBox`
  objects and, 97–98
  sorting, 79–80
Listener. *See also* trace listeners
  definition of, 615
  Output window as, 615, 616*f*
`Listeners` collection, 619
`Load` event handler, 258
Local style block
  in ASP.NET pages, 495–496
  in cascading, 501
Locals window, for debugging, 605
Logging. *See specific logs*
Loose coupling, events for, 24, 81–82

M
Machine security policy, 648–649
Managed code, *vs.* unmanaged code, 646
Managed memory, *vs.* unmanaged, 189
`ManualResetEvent`, for blocking, 173, 178
`MarshalByRefObject`, inheritance from, 239
Marshaling. *See also* Remoting
  across Web, 435
  definition of, 222, 224
  by reference, 223–239
  by value, 239–248
  into Windows Forms thread, 488–489
`Me` keyword, *vs.* `MyClass` keyword, 38–40
Member(s)
  abstract, 35
  default, 99–100
    Reflection and, 128
  definition of, 4
  encapsulation of, 27
  of interfaces, 26, 50
  nonpublic, 114
  public
    COM and, 203, 210–211
    security and, 646
  Reflection of, 101–110
  shared, 9
  surfacing, 45
  in type, 27
  virtual, 36
`MemberInfo` object, 103
Memory
  managed *vs.* unmanaged, 189
  reference types in, 10
  value types in, 10
Memory leaks, 20
Memory window, for debugging, 608
Metaclasses, definition of, 9, 27
Metadata
  in assemblies, 126, 202
  generating, 237
  server, exporting, 236
Metaphors, event handlers for, 71
Method(s)
  abstract, 60
  access modifiers for, 26
  adding, 16–22
  arguments passed to, 98
  asynchronous invocation of, 168–169
  Code Access Security for, 647
  controls bound to, 527
  debugging and, 593, 637
  definition of, 16
  dynamic changes to, 76
  factory, 522
  inherited
    calling, 18
    event handlers and, 67, 68*f*
  in interfaces, 50
  `On` prefix for, 268
  overriding, 36
  parameterized, invoking, 98
  private, definition of, 16
  proxy, Web Services and, 167, 429
  public
    constraining, 17
    definition of, 16
  Reflection of, 101–102
    invoking, 93–94
  replacing, 40–42
  virtual, 36
Method heading, *vs.* delegate, 83
`MethodInvoker` delegate, multithreading with, 85
Method signature
  definition of, 23
  event implementation with, 81
Microsoft Corporation, customer service commitment of, 187–188
Microsoft Desktop Engine (MSDE), debugging stored procedures on, 403–405
Migration, from VisualBasic 6, 673–687
  decision about, 674
  errors in, resolving, 679–685

Migration, *continued*
  of Web applications, 685–686
  of Windows applications, 675–685
Migration wizard, 674–675, 677–678, 678*f*–685*f*
Modules
  threading and, 174–175
  type and, 40
MostRecentThree method, 522
MSDE (Microsoft Desktop Engine), debugging stored procedures on, 403–405
Multicasting, by delegates, 82
Multi-language programs, debugging, 638–641
Multithreading. *See also* thread(s)
  delegates for, 85
  disadvantages of, 171
  guidelines for, 177
  Invoke method for, 163
  safety of, 173–177
  with Thread class, 177–179
  with ThreadPool class, 171–177
  with Timer class, 182–183
  Timer control and, 159–160
  in Windows Forms, 179–183
MustInherit modifier, 34–35, 148
MustOverride modifier, 148
  in abstract classes, 35
MyBase keyword, 37
MyClass keyword, *vs.* Me keyword, 38–40

N
Named arguments, 127, 143
Navigation
  automatic, Reflection for, 112
  data binding and, 283
Navigation links
  creating, 505–510, 506*f*
  test page for, 513
Nested class, 79, 417
New keyword
  object instantiation with, 18
  parameters and, 19
New operator
  in class instantiation, 9
  data passed with, 7
NewRow method, 372
Nondeterministic memory management, 19–20
NonInheritable modifier, 34
Nonpublic members, Reflection of, 105, 114
Normalization, in relational database design, 407
Nothing, structure assigned to, 8
NotOverridable modifier, 36
NT File System (NTFS), permissions in, in stored procedure debugging, 405, 405*f*
NUnit, for code testing, 354

O
Object(s)
  by reference
    vs. by-value objects, 248
    event handler attached to, 254
    for Remoting, 223–239
  by value
    vs. by-reference objects, 248
    for Remoting, 239–248
    revising for, 242–243
  caching, 518–540
  vs. class, 9
  class as (*See* Metaclasses)
  Client-Activated (CAO)
    implementation of, 237–239
    for Remoting, 234–239
    Server-Activated Objects *vs.*, 235
    server for, 235–236
  command
    DataSet filled by, 366–367
    in stored procedures, 387–388
  debugging and, 637
  definition of, 27
  fat
    DataSet as, 477
    definition of, 468
    proxy class and, 477–478
  serialization of, into DataSet, 471–474
  Server-Activated (SAO), for Remoting, 232–234
  type and, 5
  as variable, 4
Obsolete member attribute, 131, 131*f*
OleDb camp, 352–353
OleView.exe, for COM binary, 191, 192*f*
OneWayAttribute, 254, 259
On prefix, 268
Operating systems, event-driven, 64
Operations, dot operator (.) and, 9
Option Explicit On, 4
Option Strict On, 4
  late binding and, 89–90
  for type checking, 17
  type conversion and, 310
Out-of-process state server, configuration of, 534–536, 535*f*
Output parameters, for stored procedures, 394–403, 398–403
Output window
  for debugging, 611
  as listener, 615, 616*f*
Overloading
  definition of, 9
  of Dispose method, 22
  of Sub New method, 42
Overloads keyword, in Dispose method, 22
Overridable modifier, 36
  rules for, 45
Overrides modifier, 36

P
Paging, DataGrid control for, 572–573, 574*f*, 587, 587*f*
PaintEventArgs argument, delegates and, 84
PaintEventHandler class, 55, 71
  as delegate, 84
Parameter(s)
  adapter for, 368
  Reflection of, 102–103
  for stored procedures, 388
    input, 391–394
    output, 394–403, 398–403
ParameterDirection, 394
Parameterized constructor(s), 18–19
  COM and, 202
  for Remoting, 237
Parameterized methods, invoking, 98
Parent class
  constructor invocation on, 42
  definition of, 28
  methods in, calling, 37
  synonyms for, 32
Partial page caching, 539
Password, in connection string, 354

Patterns (design). *See* design patterns
Patterns (matching), regular
expressions for, 117
Performance counters, 631–634
Permissions. *See also* security
for assemblies, downloaded,
653–655, 653*f,* 655*f*
in code access security, 645
for code groups, 341, 341*f*
declarative *vs.* imperative, 655–660
granting gradually, 671
managing
installer for, 344–349
programmatically, 341–344
in runtime security policy, 652
for thin client assemblies, 335,
335*f,* 336*f*
PermitOnly action, 670
Pointer(s), reference types and, 10
Policy assemblies, in runtime
security policy, 652
Polymorphism
controversion of, 40
definition of, 28–29
enforcement of, 35
flavors of, 421
Me and, 40
MyClass and, 38
Portal modules, user controls as, 503
Positional arguments, 127, 143
Prefix notation, 4
Prime numbers, determining, 458
PrincipalPermissionAttribute,
656
PrincipalPermission class, 656
Printing permissions, demanding,
661–662
Private access, definition of, 25, 44
Private method(s), definition of, 16
Procedural types, *vs.* delegates, 82
Procedures, stored. *See* stored
procedures
Process, running, debugging,
635–636
Process class, 634
Programming tools, *vs.* user
expectations, 386–387
Project template, 411, 460–461
Properties. *See also*
DefaultPropertyAttribute
access modifiers for, 26
adding, 14–16
for binding controls, to methods,
527
binding of, to controls, 110, 111*f*
Code Access Security for, 647
of controls, applying styles with,
501–502
of custom controls
categorizing, 293–294, 294*f*
default values for, 293
editing, 288
type conversion for, 294–310
default, 319–320
definition of, 14
in interfaces, 50
adding, 54–56
overriding, 36
read-only, 15
Reflection of, 103–104
replacing, 40–42
Properties window, icon for, 293
Property keyword, 97
Protected access
definition of, 25–26, 44
generalizers and, 270
Reflecting, 101–102
Protected friend access, definition
of, 26, 44
ProvidePropertyAttribute, 313
Providers. *See* database providers
Proxy class
about, 433–435, 467–469
Begin and End methods in,
486–487
creating instance of, 168–169
DataSet and, 474–477
fat objects returned by, 477–478
generation of, 167, 428, 474–475,
474*f*
Proxy methods, 167, 429
Proxy object, in Remoting, 232, 232*f*
Proxy types, 167
Public access
in classes, 28, 43
definition of, 24–25, 44
generalizers and, 270
Public members
COM and, 203, 210–211
Reflection of, 105
security and, 646
Public methods
constraining, 17
definition of, 16
Public properties
serialized, 243
surfacing, 279–281

Q
QuickWatch window, for debugging,
606, 607*f*

R
Range checking, shadowing for, 42
Read-only properties, 15
Record lock, ADO.NET and, 352
Reference types
assignment of, 10, 12*f*
copying, 10
in memory, 10
type information for, 6
vs. value types, 5
ReflectedVersionInformation,
136–137, 141
Reflection
applications of, 110–112
for assembly attribute reading,
132–136
assembly loading with, at runtime,
91–98
of attributes, 144–150
custom, 113–114
in binding controls, 110, 111*f*,
576–577
definition of, 87
emitting dynamically with,
118–122
of events, 109–110
of fields, 104–109
late binding in, 87–90, 197
of members, 101–110
of methods, 93–94, 101–102
for navigation, automatic, 112
of objects, instance created from,
94–98
of parameters, 102–103
of properties, 103–104
security and, 114
for serialization, of typed
collections, 483
type information and, 90–91,
326
Reflection.Emit classes, 116,
120–121
regasm.exe, 204
Region directive, 40

Registration, of server
  for Client-Activated Objects, 235–236
  configuration file for, 233–234
  as service type, 230
  XML for, 233
Registry
  access to, 665–669
  type library and, 203–205
Registry read permissions, demanding, 662–663
Regular expressions
  about, 116–117
  compiled, 117–118
    emitting dynamically, 122–123
  in security, 647
  in `TextBox` control, custom, 268–270
`RejectChanges,` in `DataSet` update, 371
Relational databases
  normalization of, 407
  object-oriented systems and, 478
Relationship. *See* database relationships
Remote client. *See* client, remote
Remote events
  behavior of, 250
  handling, 249–259
  invoking, 250–259
  registration of, 256
Remote server. *See* server, remote
Remoting. *See also* client, remote; marshaling; server, remote
  about, 221–223
  application bridging with, 674
  asynchronous, 261
  with by-reference objects, 223–239
  with by-value objects, 239–248
  Client-Activated Objects for, 234–239
  definition of, 221
  event handler for, 258
  exceptions in, 259
  lifetime management in, 259
  parameterized constructors for, 237
  security of, 261
  Server-Activated Objects for, 232–234
  threads and, 259
`RemoveHandler` statement, 75
`Render` method, 321
`Repeater` control, binding data to, 569
Request-and-response model, 351
`RequestMinimum` action, 670
`RequestOptional` action, 670
`RequestRefuse` action, 670
Resource protection block (`Try...Finally` block), 363, 402
Return types, events and, 80
Reuse, information hiding and, 43
Ritchie, Dennis, 324
`RowFilter,` 373
RTTI (Run Time Type Information). *See* type information
`RunInstallerAttribute,` 345, 346
`Run` method, code for, 16
Runtime
  assembly loading at, 91–93
  emitting code at, 114–123, 115*f*
  event handler association at, 73, 75–80
  type checking at, 69–70
  type conversions at, 306–307
  type information at, 90–91
Runtime callable wrapper (RCW), 188
Runtime security policy, in Configuration utility, 651–652, 651*f*
Run Time Type Information (RTTI). *See* type information

S
Sandbox, for assembly testing, 664–665, 665*f*
SAO. *See* Server-Activated Objects
Scaffolding, definition of, 353–354
Screen layout, for ASP.NET applications, 492–503
Script, debugging, 637
Script block, for Tool Tip properties, 507
Secure HTTP, 261
Security. *See also* code access security; permissions; security policy
  assemblies and, 333–334, 333*f*
  attributes and, 157
  of binary assemblies, 470
  of connections, 671
  cookies and, 538
  declarative *vs.* imperative, 655–660
  façade for, 647
  installer file for management of, 337–349
  practices for, 644
  in programming, 646–648
  recommendations for, 670–671
  Reflection and, 114
  Remoting and, 261
  thin clients and, assemblies in, 333–334, 333*f*
  for Web applications, 546–548
  XML Web Services and, 446–456
Security attributes, interfaces and, 49
Security log, definition of, 628
Security policy
  levels of, 648–649
  managing, 648–655
  modifying, 649–653
Selector, in style block, 498
Self-testable code, 555–556
`sender` argument, 69–70
`Send` message, 254, 258–259
Sentinels, *vs.* `If` conditional statements, 91
Sentries, definition of, 16
`SerializableAttribute,` 239–242
  `ISerializable` and, 247
  public properties and, 243
  for typed collections, 483
Serialization
  automatic, 242–243
  for binding, 575
  custom, 244–247
  `DataSet` and, 419–421, 471–474
  `GetObjectData` for, 244, 485
  `IEnumerable` and, 482
  in session caching, 419, 536
  of strongly typed collections, 482–485
  XML for, 285
  XML Web Services and, 239
`SerializationInfo,` 247–248
`SerializedAt,` 248
Server, remote
  application for, 224–225, 225*f*
    configuration file for, 232–233
  channel for, 230, 256
  for Client-Activated Objects, 235–236
  client and, relationship between, 232*f*

  host for, 261
  listening by, 229–232
  metadata of, exporting, 236
  registering
    for Client-Activated Objects, 235–236
    configuration file for, 233–234
    as service type, 230
Server-Activated Objects (SAO), for Remoting, 232–234
Server controls, HTML controls as, 511–512, 512*f*
Server Explorer, SQL
  database created in, 394
  schema defined in, 412, 413*f*
Server precursors, configuration of, for thin clients, 335, 335*f*, 336*f*
Service type, server registered as, 230
Session caching
  vs. application caching, 531
  in-process state server for, 532–534
  out-of-process state server for, 534–536
  serialization for, 419, 536
  SQL server for, 536–537
Session ID, cookies and, 532, 534, 534*f*, 538
`Set` blocks, read-only and, 15–16
Set methods
  in deserialization, 248, 287
  extender providers and, 313
Setter, for property Reflection, 103–104
Setup project, defining, 347–349, 348*f*, 349*f*
Shadowing, 40–42
Shared directory. *See* Web shared directory
Shared event wrapper, implementation of, 251–256
Shared members, 9. *See also* Metaclasses
Sibling class, 32
Sieve of Eratosthenes, 458–460
Simple Object Access Protocol (SOAP), for returning data, 463
`SingleCall`, 231
`Singleton` objects, 249, 628
Sink, 198
Slicing problem, 20
Smart client. *See* thin client
SOAP (Simple Object Access Protocol), for returning data, 463
Solution Explorer, Web reference in, 429, 430*f*
Sort behavior, dynamic, 373–376
  callback function for, 76–80
Sorting
  of array, `IComparer` for, 577–579
  `DataGrid` for, 573–579, 574*f*
  of `DataSet`, 373–376
  in XSLT, 542
Source, 198
Specifications, deviation from, 276
SQL
  debugging, 637
  generating
    with command builder, 367–368
    for update, 386
SqlClient camp, 352
SQL Query Analyzer, table created with, 395–396, 395*f*
SQL server
  connection string for, defining, 390–391, 391*f*
  debugging in, of stored procedures, 406–407
  .NET debugging and, 403
  for session management, 536–537
  table defined in, 394–396
SQL Server Explorer
  database created in, 394
  schema defined in, 412, 413*f*
State variables. *See* fields
Stored procedures
  about, 387
  debugging, 403–407
    in MSDE, 403–405
    in SQL server, 406–407
  implementation of, 396–398
  input parameters for, 391–394
  invoking, 388–390
  output parameters for, 394–403, 398–403
  programming with, 387–403
  Windows Form application for, 398–403, 398*f*
`Stream` class, in trace listener initialization, 619
`StreamingContext`, 247–248
Strongly typed collection
  about, 479–482
  implementation of, 479–480, 519–522
  navigation links as, 508–510
  returning, 478–486
  serializing, 482–485
  storing, in application cache, 523–524
Structure(s)
  assignment of, 8
  definition of, 5, 6
  equality testing of, 8
  event implementation in, 23
  events in, declaring, 81
  parameterized constructors for, 19
  properties of, 6–7
  support in, 6, 13
  vs. types, 6
Style
  applying to control, 501–502
  cascading of, 501
Style block, in external style sheet, 498–499, 498*f*, 499*f*
Style Builder, 500, 500*f*
Style sheets, cascading, 494–497
Subclass. *See* child class
`Sub New` method
  implementing, 18
  overloading, 42
Superclass. *See* parent class
Surfacing constituent object members, 45
Surfacing constituent properties, 279–281
Synchronous behavior
  execution of, 161
  for file operations, 170
  programming, 161–163
`SyncLock`, for thread safety, 175–177
`System.Attribute` class, inheriting from, 143
`System.ComponentModel.Component`, inheritance from, 264
`System.Convert` class, 295
`System.Data` namespace, 353
`System.Int32`, `Integer` type and, 13
System log, definition of, 628
`System.Text.ASCIIEncoding`, 170
`System.Threading.Timer` class, 182

- `System.Type` class, binder in, 99
- `System.Xml` namespace, 353

T

- Tables. *See also* HTML tables
  - in `DataSet`, 361
  - default view from, 557–558, 558*f*
- Template(s)
  - for navigation links, 505–506, 506*f*
  - for projects, 411
  - use of, 577
  - user control as, 528, 528*f*
- Template columns, 586–588
- Temporary variables, 522
- Terrarium, 324
- Testing
  - assembly sandboxing for, 664–665, 665*f*
  - black box, 439–440
  - code for, writing, 555–556
  - of connection strings, 358–360, 555–556
  - of custom components, 270–271, 271*f*
  - custom event log for, 628–629
  - equality, of structures, 8
  - NUnit for, 354
  - XML Web Services, 438–441
- `TextBox` control
  - binding, 562–569, 563*f*
  - custom, 268–270
- `TextWriterTraceListener`, 619
  - with `FileStream` object, 620–622
- Thin client
  - configuration of, 329–335
  - definition of, 91–92
  - implementation of, 324–329
  - Internet connection for, 329–330, 337
  - in Remoting, 222
  - Trusted Sites and, 330–333, 331*f*
  - uses of, 327–329
  - Web shared directory for, 325, 327, 328*f*
- Thread(s). *See also* multithreading
  - for event handlers, 259
  - programming with, 171–179
  - Remoting and, 259
  - safety of, 173–177
  - `ThreadPool` for, 173
- `Thread` class
  - multithreading with, 177–179
  - *vs.* `ThreadPool` class, 171
- `ThreadPool` class
  - multithreading with, 171–177
  - *vs.* `Thread` class, 171
- `Tick` event, 159–160
- `Timer` class, multithreading with, 182–183
- Timer control, 159–160
  - *vs.* asynchronous processing, 170–171
- `tlbexp.exe`, 202, 204
- `tlbimp.exe`, for COM importing, 192*f*
- `ToBoolean` method, 295
- Toolbar, extender provider for, 311–314
- Toolbox
  - code added to, 385
  - control added to, 271–273, 272*f*
  - documentation added to, 277–278, 277*f*
- `ToolBoxBitmapAttribute`, 274
- Tool Tip, script block for, 507
- `ToolTip` control, 310–311
- `ToString`, 301–302
- `ToType`, 301–302
- `Trace` class, for debugging, 618–619
- Trace information
  - `EventLog` component for, 264
  - from `Send`, 254, 255*f*
- Trace listeners
  - custom, 619, 623–625
  - definition of, 619
  - event logging with, 622, 623*f*
  - initialization of, 619
  - logging with, 620–622
  - programming with, 619–625
- Transactions, in ADO.NET, 407–410
- Trust an Assembly, 333–334, 333*f*, 650
- Trusted Sites, thin clients and, 330–333, 331*f*
- `Try` block, 92–93
- `Try...Catch...Finally...EndTry` block, 409–410
- `Try...Finally` block, 363, 402
- Type
  - algorithm and, 577
  - complex
    - flattening, 468
    - implementation of, 463–466
  - emitting dynamically, Reflection for, 118–122
  - exposing to COM, 203–205
    - debugging and, 217–218
  - members in, 27
  - modules and, 40
  - `Object` class and, 5
  - properties of, 5
  - proxy, 167
  - Reflection of, instance created from, 94–98
  - type converter associated with, 306
- Type checking
  - `Option Strict On` for, 17, 89
  - runtime, `sender` argument and, 69–70
  - in type conversion, 305
- Type construct, structure and, 6
- Type conversions
  - `Binder` class for, 99
  - for custom control properties, 294–310
  - at design time, 307–310
  - exceptions in, 301
  - at runtime, 306–307
  - type checking in, 305
- Type converter
  - definition of, 295
  - existing, 306
  - implementation of, 302–306
  - type associated with, 306
- Typed collection, 479. *See also* strongly typed collection
  - `Add` method and, 485, 486*f*
  - as array, 481–482
- Type editor, for custom controls, 288–293
- `Type.GetMember` method, 103
- Type information
  - for reference types, 6
  - Reflection and, 87, 90–91
  - at runtime, 90–91
  - `TypeOf` keyword for, 91
  - for value types, 6
- Type library, 203–205
- `TypeOf` keyword, for type information, 91

U

- UDDI (Universal Description, Discovery, and Integration), 428
- `UITypeEditor` class, 288–294
- Unboxing, 10–13
- Uniform Resource Identifier (URI), for service, 230

Unit testing, self-testable code for, 555–556
Universal Description, Discovery, and Integration (UDDI), 428
UNIX, hacks and, 644
Unmanaged code, *vs.* managed code, 646
Unmanaged memory, *vs.* managed, 189
URI (Uniform Resource Identifier), for service, 230
User, in Windows authentication, defining, 449–454, 451*f*, 452*f*, 453*f*, 454*f*
User control. *See also* control
  adding to page, 510, 511*f*
  in ASP.NET, 503–516
  composite
    data binding and, 581
    defining, 579–582
    definition of, 503
    implementation of, 580–581, 580*f*
    repeating, `DataList` for, 579–585
    Web page with, 582–585, 583*f*
  converting Web page into, 514–516
  custom, 278*f*
    data binding to, 281–283
    implementation of, 278–283
    for Web, 320–321
  as `DataList` template, 528, 528*f*
  defining, 525–527, 525*f*
  dynamically loading, 510–514, 513*f*
  editable version of, 585, 586*f*
  inheritance in, 516
  layout for, 504–505
  read-only version of, 585, 586*f*
  repeating automatically, 527–530, 531*f*, 583*f*
`UserControl` object
  adaptation with, 47–48
  creating, 46–48
  definition of, 46
User expectations, *vs.* programming tools, 386–387
User interface
  Reflection for generating, 469–470, 471*f*
  of Web applications, 373
User interface generator, 141, 144–150
User security policy, 648

V
Validation, regular expressions for, 268
Value types
  assignment of, 10, 11*f*
  copying, 10
  definition of, 5
  in memory, 10
  type information for, 6, 10 (*See also* boxing)
Variables
  declaring, 3–5
  state (*See* fields)
  temporary, 522
  thread safety and, 173–177
VCL (Visual Control Library), 222
Verb, 314–317
View state, 538–539
Virtual abstract classes, 107
Virtual member, `Overrides` modifier for, 36
Virtual methods, 36
Virtual methods table, 29
VisualBasic 6
  attributes in, 66
  consuming assemblies in, 205–207, 206*f*
  event handler in, 66
  features of, not supported in .NET, 675, 676*t*
  migration from, 673–687
Visual Control Library (VCL), 222

W
Wahoo!, 331
`WaitCallback` delegate, multithreading with, 85
`WaitHandle` objects, blocking with, 166–167
Watch window, for debugging, 605, 606–607, 608*f*
Web application
  debugging, 636–638
  migration of, from VisualBasic 6, 685–686
  security for, 546–548
  user interface of, 373
  vs. Windows application, 323
Web application director, creation of, 443–444, 443*f*
`Web.config` file
  authentication in, 446, 546–548
  authorization in, 454–456
  connection string in, 552–553
  default, 445–446, 446–448
Web controls
  custom, 320–321
  multi-value, binding data to, 569–571
  single-value, binding data to, 562–569
Web Forms controls, view state of, 538–539
Web methods
  asynchronous invocation of, 488
  `Begin` method in, 486–487
  `DataSet` returned by, proxy class for, 474–477
  `End` method in, 486–487
  writing, 438
Web page
  caching, 539–540
  with composite user control, 582–585, 583*f*
  converting user control into, 514–516
  `IsPostBack` property of, 588
Web reference, 466–467, 467*f*
  adding, 428, 429*f*, 430–431
  in Solution Explorer, 429, 430*f*
Web Service
  definition of, 167
  file components of, 441, 444
  XML (*See* XML Web Service)
`WebService` attribute, applying, 437
Web Service class, declaring instance of, 431–432
Web Service methods
  asynchronous calling of, 433
  invoking, 432
Web Services Description Language. *See* WSDL
Web shared directory
  creating, 335, 335*f*, 336*f*
  for thin client application, 325, 327, 328*f*
Web user controls, custom, 320–321
Windows application
  debugging, 636–637
  migration of, from VisualBasic 6, 675–685
  vs. Web application, 323
Windows authentication, with Web Services, 449–456, 449*f*, 450*f*

- Windows control, custom, 267–271
  - adding to Toolbox, 271–273, 272*f*, 273*f*, 274*f*
  - bitmap for, 273–276, 275*f*, 276*f*
  - designer for, 314–319, 317*f*, 318*f*
  - documentation for, 277–278, 277*f*
  - inheritance of, 267
  - testing, 270–271, 271*f*
- Windows Forms
  - inheritance in, 33
  - multithreading in, 179–183
  - for stored procedure, 398–403, 398*f*
- Windows Forms application, for thin client, 325–326
- Windows Forms authentication, for Web applications, 546–548
- Windows Forms thread, marshaling data onto, 488–489
- Windows Installer package, policy settings modified with, 650
- Windows operating system, events in, 63
- `WindowsPrincipal` class, 658
- Windows security, Code Access Security and, 645–646
- `WithEvents` statement, event handlers and, 72–73
- Wizards
  - migration with, 674–675, 677–678, 678*f*–685*f*
  - policy management with, 650
- World Wide Web, simultaneous connections on, 351
- Wrapper classes, for Web Service methods, 429
- `WriteEntry` method, versions of, 9
- WSDL (Web Services Description Language), 167, 428
  - CodeDOM and, 469
  - proxy file generated by, 467–469
- WSDL utility
  - `Begin` method and, 486
  - `End` method and, 486

X

- XML (eXtended Markup Language)
  - as data, 541–542
  - dynamic interfaces with, 540–546
  - for serialization, 285
    - of typed collection, 482–483
  - for server registration, 233
- XML document
  - caching dependency for, 545–546
  - style sheet for (*See* XSLT)
- XML Schema
  - adding to project, 412
  - defining, 411–414
  - relationships added to, 414, 415*f*
- XML Schemas and Data (XSD), about, 411–412
- XML Style Transform. *See* XSLT
- XML Web Service
  - application bridging with, 674
  - asynchronous invocation of, 167–169, 486–489
  - collection returned from, 485
  - consumer of, implementation of, 469–471
  - creating, 436–438, 436*f*
  - data returned from, 458–462
    - complex, 463–471
  - `DataSet`s returned from, 419, 471–477
  - debugging, 438–441
  - deployment of, 441–446, 442*f*
  - existing, 428–435
  - fat objects and, 468–469
  - finding, 427–428
  - implementation of, 460–462, 462*f*, 466
  - referencing, 466–467, 467*f*
  - security and, 446–456
  - serialized data and, 239
  - Sieve of Eratosthenes as, 459–460
  - testing, 438–441
  - Windows authentication and, 449–456, 449*f*, 450*f*
- XSD (XML Schemas and Data), about, 411–412
- XSLT (XML Style Transform)
  - caching dependency and, 545–546
  - implementation of, 542–545, 544
  - for layout description, 540–541