

Index

Symbols

\<, 261
 \>, 261
 \b, 261
 \B, 261
 \d, 261
 \w, 261
 \s, 261
 \n, 246
 \r, 246
 \r\n, 246
 \“, 246
 \\, 246
 \t, 246
 {n}, 261
 {n, }, 261
 {n, N }, 261
 * /, 74
 / *, 74
 //, 74
 < >, 82
&& (AND), 106, 107
+ (addition operator), 108, 279
 overloading, 196
 string concatenation with, 250, 257
= (assignment operator), 103, 106
 delegates and, 220, 235, 240
/ (division operator), 108
== (equal), 105, 248
> (greater than), 105
>= (greater than or equal to), 105
< (less than), 105
<= (less than or equal to), 105
% (modulus operator), 108, 109
***** (multiplication operator), 108
! (NOT), 106, 107
!= (not equal), 105
?? (null coalescing operator), 83
\$ (regex operator), 261
***** (regex operator), 261
+ (regex operator), 261
. (regex operator), 261
? (regex operator), 261
^ (regex operator), 261
- (subtraction operator), 108

_ (underscore), 133
?: (conditional operator), 93, 109
|| (OR), 106, 107
[] (regex operator), 261
[^] (regex operator), 261
+ = operator, 220, 235, 237
- = operator, 220
@ character, 70, 246, 247, 749
(), forcing precedence with, 110
: (colon) operator, 185
, (comma) separator, 400

A

ABCs of WCF, 710–711. See also WCF
Abort () method, 297, 298
aborting threads, 297–301
abstract base classes, 171, 186
 interfaces *v.*, 171, 211
abstract keyword, 69, 186, 195, 750
abstract methods, 186–189
access modifiers, 133, 200–201
 internal, 133, 200–201
 private, 133, 142, 146
 protected, 133, 200–201
 public, 133
accessors
 get, 140, 141, 142, 146, 749
 set, 140, 141, 142, 146, 749
Active Server Pages .NET. See ASP.NET
ActiveSync, 596, 598, 603, 604, 606,
 612, 614
Add () method, 214, 215, 405, 420
Add New Item dialog, 605, 742
Add New Smart Device Project dialog, 579
Add Reference dialog, 470, 757
Add Service Reference dialog, 689, 697, 698, 708,
 718, 727
Add Silverlight Link, 691, 692
Add to Reference button, 764
Add Watch feature, 52
Add Web Reference dialog, 698, 719
AddContact () method, 138
addition operator (+), 108, 279
 overloading, 196
 string concatenation with, 250, 257

AddNumbers () method

AddNumbers () method, 135, 136

addresses (WCF service), 710, 711

AddTwoNumbers () function, 221, 222, 224

Administrator (Vista), 493, 495

ADO.NET, 425, 438, 758. See also pubs sample database

LINQ to DataSet and, 438

pubs database and, 438

AfterInstall event, 606

AfterInstall event handler, 608

Age () method, 180, 181

Aggregate () function, 443

aggregate functions, 435, 442–443

Aggregation operators, 432

AJAX, 560, 617, 759

Control Toolkit, 561–562

definition of, 560

Extender controls, 561, 562

Extensions, 560

ModalPopupExtender control, 566–572

ScriptManager control, 563

Silverlight v., 675

UpdatePanel control, 563–564

UpdateProgress control, 565–566

WCF services and, 741–745

AjaxControlToolkit–Framework3.5–NoSource.zip, 561, 562

AjaxControlToolkit–Framework3.5.zip, 561

AJAX–enabled WCF Service template, 741, 743

AJAX–enabling an ASP.NET web page, 563

AlarmClock class, 232, 233, 234, 235, 240

AlarmClockEventHandler, 238

AlarmTime property, 233, 235

algorithms. See hashing algorithms

alias, namespace, 485

AllNumbersZeroException class, 393, 394, 395

Allow Anonymous Connections checkbox, 505

Alphabetical view, 28

anchor points, 24

AND operator (&&), 106, 107

Animations project, 645–650

Animations2 project, 650–654

anonymous methods, 225–226

anonymous types, 78, 128–130

immutable, 129

LINQ and, 436–437

LINQ to DataSet and, 441–442

literals and, 130

app.config file, 512

Append () method, 258, 259

AppendAllText () method, 327

AppendFormat () method, 259

AppendLine () method, 259

application development

ASP.NET web applications, 545–572

Silverlight applications, 617–694

WCF and, 695–745

Windows applications, 503–543

Windows Mobile applications, 573–616

Application Downloading Complete message, 613

Application Files folder, 541, 542

Application Install dialog–Security Warning, 536

application manifest (PhotoViewer), 541–542

application settings (PhotoViewer), 508–513

default values for, 510

definitions for, 510–512

FTP_SERVER, 509

Password, 509

purpose of, 509

scope for, 510

types for, 510

UserName, 509

application testing. See unit tests

ApplicationDeployment static class, 540

applications, service oriented, 695. See also WCF

architecture

LINQ, 425–426

Silverlight, 620–621

Area () method, 187, 189, 192, 207

AreEqual () method, 56, 58, 59

args parameter, 68

ArgumentException error, 420

arguments

parameters v., 135

passing, into methods, 135–137

ArithmeticException exception, 383, 384, 385

Array base class, 399

array initializer, 402

ArrayList class, 405–407

arrays, 87–88, 397–404

arrays of, 402–403

copying, 404

defined, 397

dynamic, 405–407

elements, accessing, 400

initialization of, 398–399

of integer values, 429

jagged, 402–403

lines, 402

multidimensional, 400–401

num, 398, 399, 404

parameter, 403–404

Point, 402

rank specifier in, 398

sorting, 399–400

string, 68, 427, 428, 429, 430

as keyword, 69, 161, 750
as operator, 178–179
AsDataView () method, 441
ASMX Web Services. *See* **Web Services**
ASP, classic, 546
.asp extension, 546
ASP.NET, 758
 AJAX. *See* **AJAX**
 .aspx extension and, 546
 data binding, 547–548
 GridView control and, 549–554
 history, 545–546
 IIS and. *See* **IIS**
 LinqDataSource control, 547, 550, 555, 556, 558, 572
 requirements for, 546–547
 Silverlight v., 675
 web applications, 545–572
 Web Development Server, 696–697, 714
 web page, AJAX-enabling, 563
.aspx extension, 546. *See also* **Default.aspx**
assemblies (.NET), 6–8, 467–499
 contents of, 468–469
 defined, 467
 directory locations of, 758
 in GAC, 486, 492–494
 library, 467
 manifest of, 469
 metadata, 467, 468
 modules and, 468, 474–480
 MSIL and, 6–8, 468
 multi-file, 469–480
 namespaces and, 480–486
 Object Browser and, 763–764
 parts, 467
 private, 486
 process, 467
 resources, 467
 shared, 486–492
 single-file, 469
 type metadata, 467, 468
 versioning, 490–491, 492, 494
Assembly Cache Viewer, 493
AssemblyFileVersion attribute, 491, 494
AssemblyInfo.cs file, 490, 491, 493
AssemblyVersion attribute, 491, 493, 494
assignment operators, 103–105, 109
asymmetric encryption, 350–353, 487
AsymmetricDecryption () function, 352
AsymmetricEncryption () function, 351
AsyncCallback delegate, 224
asynchronous callbacks, 222–225
 threading and, 225
Asynchronous JavaScript, 560. *See also* **AJAX**
at (@) character, 70, 246, 247, 749

atomic increment/decrement operations, 304
attributes, 162–165. *See also* **specific attributes**
 applied to classes, 164
 CLR, 163–164
 custom, 164–165
AttributeUsage attribute, 165
authors table, 438, 439, 440, 458, 459
Auto Hide button, 24, 25
automatic properties feature, 145–146, 173
automobile computing devices, 574
auto-play (Media project), 657–658
Autos window, 53
Average () function, 443

B

BackgroundCssClass attribute, 571
BackgroundWorker control, 293, 315–319
bare-bones application. *See* **Silverlight**
base class, 184. *See also* **abstract base classes**;
 specific base classes
base class constructors, calling, 203–204
Base Class Library (BCL), 5, 758, 759
base keyword, 69, 187, 188, 203, 750
Base64 encoding, 488
Base64Codec, 488, 491, 495, 496
Base64Codec.dll assembly, 494
BASIC endpoint, 715, 716, 717
BasicHttpBinding, 712, 714, 716, 717, 719, 720
BCL. *See* **Base Class Library**
BeginInvoke () method, 224, 312
BeginPrint event, 527, 529
BeginRead () method, 335
binary serialization, 359–365
BinaryReader class, 329
BinaryWriter class, 329
bindings (WCF), 710, 711, 712
 BasicHttpBinding, 712, 714, 716, 717, 719, 720
 MexHttpBinding, 712, 717
 MsmqIntegrationBinding, 712
 NetMsmqBinding, 712
 NetNamedPipeBinding, 712
 NetPeerTcpBinding, 712
 NetTcpBinding, 712, 714, 724, 725, 735
 WSDualHttpBinding, 712
 WSFederationHttpBinding, 712
 WSHttpBinding, 706, 712, 714, 716, 719, 720
booking tickets. *See* **WCF ticketing service**
Bookmark class, 359, 360, 362, 363
bool (keyword), 69, 750
bool (value type), 80
Branch class, 289, 290, 291
BranchNamesCollection class, 291
break keyword, 69, 94, 95, 99–100, 750

breakpoints, 49–50

browsers

- Firefox, 560, 618
- Internet Explorer, 536, 554, 560, 618, 623
- Safari, 618
- Silverlight and, 618

Brush Transform tool, 639

btnCreateFolder, 506

btnDeletePhoto, 506

btnPreview control, 527

btnPrint control, 527

btnRemoveFolder, 506

btnUploadPhotos, 506

BufferedStream class, 331, 333–334

Build the Help File button, 776

BuildDirectory () function, 516, 519

BuildReflectionData.bat batch file, 774

buttons. *See specific buttons*

byte (keyword), 69, 750

byte (value type), 80

- conversion, 89
- range, 80

byteArrayToString () function, 357

C

C# compiler (csc.exe), 64–66

C# language

- application development with. *See application development*
- foundations, 61–123
- object-oriented programming. *See object-oriented programming*
- strongly typed, 78, 89, 280, 281, 447
- syntax, 68–123

Ⓢ tag, 768, 771

CAB (cabinet) file, 598–603

cabinet (CAB) file, 598–603

Calculation class, 209, 210

callback delegate, 221, 222

CallbackContract property, 731

CallbackFunction () function, 744

callbacks, 728

- asynchronous, 222–225
- defined, 220, 728
- implementing
 - delegates and, 220–222
 - WCF, 728–741
- synchronous, 222

calling base class constructors, 203–204

calling WCF services, 710

calling Web Services, 293, 710

CallMathUtil () function, 478, 479

CallMathUtil () method, 472, 473

CallStringUtil () function, 478, 479

CancelAsync () method, 319

CancelControlID attribute, 571

CanSeek (), 338

Canvas control, 624–626, 628, 635, 636, 640, 641, 644, 659

capture of signature, 677–678

cascade–delete operations, 464

case keyword, 69, 750

case, string, 254

casting

- to interfaces, 177–178
- type, 90
- type conversion, 89–92

catch keyword, 69, 750

CEAPPMGR.EXE, 606

cgacutil.exe utility, 577

chaining

- constructor, 149–151
- delegates, 217–220
- exception, 387–389

char (keyword), 69, 750

char (value type), 80

- conversion, 89
- range, 80

checked keyword, 69, 90, 750

CheckForUpdate () method, 540

.chm (MSDN–style HTML Help format), 773, 780

Choose Toolbox Items dialog, 23

cinema ticketing service.

See WCF ticketing service

circle, circumference of, 189

Circle class, 189, 190, 204

circularly linked lists, 284

circumference, of circle, 189

class keyword, 69, 126, 750

class libraries (.NET Framework), 4–5, 66

- BCL, 5, 758, 759
- FCL, 5
- generics and, 280–284
- Object Browser and, 759–764

Class Library template, 604

Class2.cs, 481

classes, 66, 125–126. *See also specific classes*

- abstract. *See abstract base classes*
- attributes applied to, 164
- base, 184
- Contact. *See Contact class*
- contracts for, 171, 181. *See also interfaces*
- defining, 126
- derived, 184
- generic, 268–269
 - LinkedList Ⓢ, 284–287
- implementing, 171, 173–175, 181
- instance of, 127–128. *See also instantiation*

- members. See members
- sealed, 191–192
- static, 156–157
- structs v., 165–166, 169
- in System.Collections.Generic namespace, 280–281
- as templates, 125, 127
- classic ASP, 546. See also ASP.NET**
- ClassLibrary1 project, 480**
- ClassLibrary1.dll, 481**
- ClassLibrary2 project, 481, 482**
- ClassLibrary3 project, 482, 483, 484**
- ClassLibrary3.dll, 484**
- clean as you go, 765. See also documentation**
- ClearItems () method, 290**
- Click event, 228**
- ClickOnce, 503, 534–536**
- ClientBin folder, 692**
- clients, 171**
 - smart, 534
 - for WCF ticketing service, 736–740
- client–server socket application, 338–340**
- CLR (Common Language Runtime), 3–4**
 - attributes, 163–164
 - execution engine, 621
 - JIT compiler, 4, 9, 467, 499
 - object destroying and, 154
- coalescing operator, null, 83**
- code**
 - documenting, 74–78, 765–780
 - locked, 305
 - managed, 3
 - refactoring, 39–48
 - reuse, generics and, 271
 - stepping through, 51
 - unmanaged, 3, 4, 154, 155
 - view, 32–34
- Code and Text Editor, 34–39**
- Code Editor, 35, 37, 49, 120, 121, 563**
- code generator, 144**
- Code Snippet feature, 35–37**
- code tag, 768, 771**
- code-behind, 30, 31**
- Collapse All Feeds Menultem control, 596**
- CollapseAll () method, 596**
- collection classes, 397**
- collections interfaces, 397, 404–405.**
 - See also arrays
- Collection{T} class, 288, 289, 290, 291**
 - List{T} class v., 291
- colon (:) operator, 185**
- COM objects, 23, 154, 155**
- comma separator (,), 400**
- comments, 74**
- Common Language Runtime. See CLR**
- Common Type System (CTS), 85**
- communication models. See also callbacks**
 - one-way (simplex), 700, 713, 723, 731
 - request/response, 700, 713, 728
 - two-way (duplex), 700, 713
- communication protocols**
 - HTTP, 534, 546, 547, 560, 618, 700, 710, 711, 712
 - MSMQ, 710, 712
 - named pipes, 710, 711, 712
 - SOAP, 618, 699, 700, 706, 712
 - TCP, 331, 338, 339, 710, 711, 726
 - WCF, 710
- Compare () method, 418**
- CompareTo () instance method, 249**
- CompareTo () method, 273, 415, 416, 417**
- comparing strings, 248–249**
- CompositeType class, 703, 707**
- Compress () function, 353, 354, 356**
- compressions, 353–354**
- concatenating strings, 250–251**
 - StringBuilder class and, 257–259
- ConcurrencyMode property, 734**
- conditional operator (? :), 93, 109**
- Configuration Editor, WCF Service, 714, 724**
- Configure Where Expression dialog, 557**
- conn object, 155**
- Console.WriteLine () method, 30, 254**
- const keyword, 69, 750**
- constants, 74**
- constraints, in generic types, 271–275**
- constructed types, 269**
- constructor chaining, 149–151**
- constructors, 146–165**
 - base class, calling, 203–204
 - copy, 152–153
 - default, 147
 - structs and, 166
 - XML serialization and, 374–375
 - inheritance and, 202
 - object initializers v., 154
 - static, 151–152
- ConsumeWCFService, 708**
- Contact class, 126**
 - access modifiers in, 133
 - automatic properties, 145–146
 - CLR attributes and, 163–164
 - constructors and, 147, 148, 149
 - copy constructor in, 152–153
 - instance members, 131
 - instance of, 127–128
 - instances, testing for equality in, 158–159
 - LINQ and, 436–437
 - members, 126, 131
 - object initializers and, 153–154
 - Obsolete attribute applied to, 164
 - partial classes, 126–127

Contact class (continued)

- partial methods, 143–145
- properties and, 139–140
- static constructors and, 151–152
- static member, 132
- ToString () method and, 161–162

Contains () method, 406

ContainsKey () method, 420

contextual keywords, 749–750, 754–755

continue keyword, 69, 102, 750

contracts

- for classes, 171, 181. *See also* interfaces
- WCF, 710, 711, 712
- data, 712
- fault, 712
- message, 712
- operation, 712
- service, 712

controls. *See also specific controls*

- missing, 25
- for PhotoViewer, 506–508
- Toolbox and, 22–25

Conversion operators, 432

conversions. *See* casting

ConvertToTimeSpan () function, 669–670

CoolLabs.net namespace, 482, 485

Coordinate struct, 166, 167, 168, 276

copy constructor, 152–153

Copy () instance method, 251

Copy Local property, 471, 496

Copy () method, 327, 404

copying arrays, 404

CopyTo () method, 251, 259

CopyToDataTable () method, 449, 450

Count () function, 435, 442, 443

Count () method, 429

count static member, 132

counter, 258

Cradle, 596, 597

crashing, 377

Create Folder button, 522

Create () method, 322, 327, 518

Create Strong Name Key dialog, 490

Create Unit Tests, 54–56

CreateDirectory () method, 324

CreatePanel () function, 584

CreateSubdirectory () method, 322

CreationTime property, 322, 323

·creator element, 456

Credit () function, 305, 310

critical section, 305, 308, 320

cryptography, 487

cryptography services, 343

- encryption/decryption, 347–353
- hashing, 344–347

- random-number generation, 285, 286, 287, 343, 346, 348, 349

CryptoStream class, 331, 348–349

csc.exe. *See* C# compiler

CSS, 560, 567, 571

CTS (Common Type System), 85

Current property, 412

current scope, 74

currentDeployment property, 540

Custom Actions tab, 610, 611

custom attributes, 164–165

custom exceptions, 393–395

custom installer (for RSS Reader), 604–608

custom stacks, 255–266

Customers table, 444, 445, 446, 449

Customize dialog

- Menu bar, 18
- Toolbar, 20–21

customizing XML serialization, 372–374

D

DAM (digital asset management) cataloging program, 620

data binding, 547–548

- GridView control and, 549–554

data contract, 712

data members, 131

Data property, 379

data reshaping, 441–442

Data Source Configuration Wizard, 550

data types, 78–85. *See also* anonymous types; reference types; typing; user-defined data types; value types

databases

- modeling, 548–549
- Northwind, 439, 443, 444, 447
- pubs, 438, 439, 458, 548, 549, 572

DataClasses1DataContext class, 460

DataClasses1.dbml, 458, 459

DataClassesDataContext object, 549, 550, 554

DataClasses.dbml, 548, 549

DataContract attribute, 703

DataSets, 438

DataTables, 438

- queries saved to, 449–450

dateCreated variable, 365

DateTime class, 457

Debit () function, 305

DEBUG symbol, 113, 117

debugging, 49

- breakpoints and, 49–50
- multiple projects, 27

decimal (keyword), 69, 751

decimal (value type), 80

Decode () method, 488

Decompress () function, 354, 355, 356

- decompressions, 354–358**
- Decrement () method, 304**
- decryption, 347–353**
- deep serialization, 364**
- default constructors, 147**
 - structs and, 166
 - XML serialization and, 374–375
- Default FTP Site item, 505**
- default keyword, 69, 82, 94, 398, 751**
 - in generics, 270–271
- default value, of value types, 81–82**
- Default.aspx, 549, 552, 555, 556, 559, 563, 565, 567, 568, 571, 744**
- deferred query execution, 432–434**
- #define, 111–115**
- /define compiler option, 114, 117**
- DeflateStream class, 353**
- delegate keyword, 69, 215, 751**
- delegates, 213–226**
 - = operator and, 220, 235, 240
 - callbacks implemented with, 220–222
 - creating, 214–216
 - defined, 213
 - events v., 235–237, 240
 - generic, 280
 - multicast, 217–220
 - ParameterizedThreadStart, 301, 302, 303
 - ThreadStart, 295, 301, 302
- delegates chaining, 217–220**
- Delete () method, 322, 324, 327**
- DeleteOnSubmit () method, 463**
- delete.png, 569**
- delta, 58, 59**
- deployment**
 - PhotoViewer, 533–543
 - RSS Reader application, 598
 - CAB file, 598–603
 - target device prerequisites, 615–616
 - shared assemblies, 488
- deployment manifest (PhotoViewer), 542–543**
- derived class, 184**
- DerivedClass, 194**
- DES algorithm, 347**
- deserialization, 370–371**
- DeSerialize () function, 362**
- Deserialize () method, 370**
- Designer window, 17, 30–31**
- destructors, 154–155**
- detecting null fields, 449**
- development settings, 12–14**
- Diagonal () method, 191, 198, 199**
- dialog**
 - Add New Item, 605, 742
 - Add New Smart Device Project, 579
 - Add Reference, 470, 757
 - Add Service Reference, 689, 697, 698, 708, 718, 727
 - Add Web Reference, 698, 719
 - Application Install dialog–Security Warning, 536
 - Choose Toolbox Items, 23
 - Configure Where Expression, 557
 - Customize
 - Menu bar, 18
 - Toolbar, 20–21
 - Edit Custom Component Set, 761
 - File Download dialog–Security Warning, 536
 - Internet Explorer dialog–Security Warning, 536
 - New Project, 15, 16, 635, 676
 - Rearrange Commands
 - Menu bar, 18–19
 - Toolbar, 21
 - Service Reference Settings, 718
 - Windows Security Alert, 726
- dictionary, 420**
- Dictionary <K, T> class, 283, 284, 420, 422, 734**
- Dictionary object, 420–422**
- digital asset management (DAM) cataloging program, 620**
- <DIR>, 519**
- directories, 321–327**
 - files and, 321–327
 - manipulating, 321–327
- directory (PhotoViewer)**
 - creating, 522
 - removing, 523
 - tree, 515–521
- Directory class, 324–325**
 - DirectoryInfo class v., 325
 - example, 324–325
 - methods, 324
- DirectoryInfo class, 322–324**
 - Directory class v., 325
 - example, 323–324
 - methods, 322
 - properties, 322
- Disassembler, MSIL, 7, 468, 491**
- Discover button, 698, 708**
- disk I/O, 353, 375**
- DisplayCurrentPlayBack () function, 669–671**
- displaying publisher's name, 554–555**
- DisplayMessage () function, 722**
- displayPopup () function, 568**
- Dispose () method, 154, 155**
- DistanceFromOrigin () method, 196**
- div element, 569, 570**
- DivideByZeroException, 381, 382, 384, 385, 386**
- division operator (/), 108**
- DLL Hell, 486**
- DLLs (library assemblies), 467**
- Dns class, 585**
- do keyword, 69, 751**

documentation (of code), 765–780

- building/viewing, 776–779
- distributing, 779–780
- generating, 773
- Sandcastle tool, 78, 765, 773
 - downloading, 774
 - installing, 774
 - launching, 775–776
- XML, 74–78, 765–773
- DoSomething () function, 294, 295, 296, 297, 299, 300, 301, 302, 303**
- double (keyword), 69, 751**
- double (value type), 80**
- doubly linked lists, 281, 284**
- do-while loop, 99**
- DownloadFeed () function, 585**
- DownloadImage () function, 520, 521**
- DownloadProgressChanged(), 667, 671**
- DoWork event handler, 317, 318**
- DoWorkEventArgs, 318**
- DrawSignature () subroutine, 684**
- DropDownList control, 556**
- DSA algorithm, 351**
- duplex communication model, 700, 713**
- dynamic arrays, 405–407**

E

- Edit Custom Component Set dialog, 761**
- Edit WCF Configuration, 714, 735**
- e.Graphics.DrawRectangle () method, 529**
- e.Graphics.DrawString () method, 529**
- Elapsed event, 229, 230, 233**
- Element operators, 432**
- elements. See *specific elements***
- #elif, 115–117**
- ‘Ellipse’ element, 624, 625, 629**
- Ellipse object (RoundButton project), 636, 637, 638, 639, 640**
- EllMarkerMouseDown (), 667, 671**
- EllMarkerMouseUp (), 667, 671**
- #else, 115–117**
- else keyword, 69, 751**
- email validation, with regex, 263**
- Employee class, 174, 181, 183, 184, 274, 275, 414**
- emulators, Windows Mobile, 577, 596–597**
- Encapsulate Field, 44–46**
- encapsulation, 139**
- Encode () method, 488**
- encryption, 347–353**
 - asymmetric, 350–353, 487
 - defined, 347
 - hashing v., 347
 - private key, 347, 487
 - public key, 350, 487
 - symmetric, 347–350, 487
- #endif, 115–117**
- EndInvoke () method, 312**
- endpoints, 700, 711**
 - BASIC, 715, 716, 717
 - multiple, WCF service and, 714–720
 - WS, 714, 716
- Endpoints node, 714, 715**
- EndPrint event, 527, 529**
- EndRead () method, 335**
- #endregion, 120–122**
- EndsWith () instance method, 253**
- Enqueue () method, 424**
- Enter () method, 308**
- entry point, 67**
- enum keyword, 69, 85, 751**
- enumerations, 85–87**
- environment settings, 12–13**
- equality**
 - reference, 157, 159
 - testing for, 158–159
 - strings, 248
 - value, 159
- Equality operators (LINQ), 432**
- equals, implementing, 159–161**
- Equals () method, 157, 158, 159, 160, 161, 248**
- equals sign (=)**
 - = (assignment operator), 103, 106
 - delegates and, 220, 235, 240
 - == (equal), 105, 248
 - >= (greater than or equal to), 105
 - <= (less than or equal to), 105
 - != (not equal), 105
- #error, 117–118**
- Error List window, 17, 29–30**
- escape characters, 243–247**
 - list, 246
 - Unicode and, 247
- event handlers, 229–232. See also *specific event handlers***
 - defined, 227
 - MediaPlayer project, 667, 671–675
 - RSS Reader application and, 589–596
 - state information passed to, 237–240
- event keyword, 69, 751**
- EventArgs class, 228, 233**
- event-driven programming, 226**
- EventHandler, 233**
- events, 226–240. See also *specific events***
 - defined, 213, 227
 - delegates v., 235–237, 240
 - handling, 229–232
 - implementing, 232–235

- Events icon, 29**
 - example tag, 768, 771**
 - exception chaining, 387–389**
 - Exception class, 379, 380**
 - properties, 379
 - exception objects, 389–391**
 - exception tag, 768**
 - exceptions**
 - custom, 393–395
 - defined, 377
 - examples of, 377
 - handling, 377–395
 - try-catch statement and, 378–381
 - inner, 387
 - multiple, handling, 381–383
 - rethrowing, 386–387
 - ThreadAbortException, 298
 - throwing, 383–387
 - EXE, 467. See also assemblies**
 - executables, 713**
 - Exists () method, 324, 327**
 - Exists property, 322, 323**
 - Exit () method, 308**
 - exiting from loops, 99–102**
 - explicit interface members implementation, 208–211**
 - explicit keyword, 69, 751**
 - explicit type conversion, 90–92**
 - explicit typing, 88**
 - Expression Blend 2, 620, 635–644**
 - RoundButton project and, 635–644
 - trial download, 635
 - Expression Design, 620**
 - Expression Media Encoder Preview Update, 620**
 - Extensible Application Markup Language (XAML).**
 - See XAML
 - extension methods, 198–199**
 - LINQ and, 430–436
 - extern keyword, 69, 195, 751**
 - Extract Interface feature, 46–48**
 - Extract Method feature, 40–42**
- F**
- fall, rate of, 653–654**
 - fall-throughs, 95**
 - false keyword, 69, 751**
 - fault contract, 712**
 - FCL. See Framework Class Library**
 - Fibonacci () method, 470, 474**
 - Field () extension method, 446, 447, 448, 449**
 - FIFO (first in, first out) data structure, 423**
 - files**
 - defined, 321
 - directories and, 321–327
 - manipulating, 321–327
 - reading/writing to, 327–328
 - streams v., 321
 - File class, 326–327**
 - File Download dialog–Security Warning, 536**
 - File Explorer, 577, 602**
 - file explorer, creating, 329–331**
 - file I/O, 312, 337, 375**
 - FileInfo class, 326–327**
 - FileStream class, 331, 334–337**
 - FilesUtil class, 156**
 - FileSystemWatcher class, 230, 231**
 - Fill attribute, 630, 631**
 - finally keyword, 69, 751**
 - finally statement, 298, 391–393**
 - Find () method, 271, 272, 273**
 - findName () method, 658, 663**
 - Firefox browser, 560, 618**
 - first in, first out (FIFO) data structure, 423**
 - First () method, 463**
 - fixed keyword, 69, 751**
 - flattening data. See serialization**
 - float (keyword), 69, 751**
 - float (value type), 80**
 - conversion, 89
 - range, 80
 - floating point numbers, 58–59**
 - converting, 91–92
 - flow control, 92–95**
 - for keyword, 69, 751**
 - for loop, 96–98**
 - forced immediate query execution, 434–436**
 - forcing precedence, 110**
 - foreach keyword, 69, 751**
 - foreach loop, 98, 400, 433, 437, 441, 442**
 - Form1_Load event handler, 313, 314, 709**
 - Format () method, 250, 255**
 - format specifier, 255**
 - FormatException, 381, 382**
 - formatting**
 - numbers, 255, 256, 257
 - strings, 254–257
 - Forms. See Windows Forms**
 - foundNames variable, 42**
 - Framework Class Library (FCL), 5**
 - from keyword, 69, 754**
 - from-where-select format, 428**
 - fs.Close () statement, 392**
 - FTP Server**
 - configuration of, 504–506
 - deleting photos, 525–526
 - information for access to, 508–509
 - uploading photos to, 524–525
 - FTP service, 504–506**
 - FTP_SERVER, 509**

FtpWebRequest, 503, 518
FtpWebResponse, 503, 518, 521
FullName property, 322, 323
function members, 131, 134–155
functions, 67. *See also specific functions*
 aggregate (LINQ), 435, 442–443
 methods v., 134
 pointers to. *See delegates*
 subroutines v., 67

G

GAC. *See Global Assembly Cache*
gacutil.exe utility, 493, 577
garbage collector, 154, 155, 751
General Packet Radio Service (GPRS)
 connections, 353
generated code, 144
GenerateIV () method, 349
Generation operators, 432
generic classes, 268–269. *See also specific generic classes*
 LinkedList `T`, 284–287
 LinkedListNode `T`, 285
generic delegates, 280
generic interfaces, 275–276
generic methods, 276–278
generic operators, 278–279
generic type interference, 277
generic types, 268
 constraints in, 271–275
 multiple type parameters in, 275
generics, 265–291, 405
 advantages of, 271
 code reuse and, 271
 collections interfaces and, 405
 default keyword in, 270–271
 defined, 265
 MSIL and, 265
 .NET Framework class library and, 280–284
 process of, 265–267
 type safety and, 271, 410
get accessor, 140, 141, 142, 146, 749
get keyword, 69, 749, 750, 754
getAge () function, 706, 707
GetCurrentDirectory () method, 324
GetData () method, 704
GetDataUsingDataContract () operation, 705
GetDate () method, 137
GetDirectories () method, 322, 324
GetDirectoryListing () function, 516, 518
GetEnumerator () method, 409, 410, 411, 431
GetFiles () method, 322, 324
GetHashCode () method, 157

GetHostEntry () method, 585
GetHostName () method, 585
GetIniPath () function, 607
GetLowerBound () method, 401
GetParent () method, 325
GetResponse () method, 518
GetResponseStream () method, 518, 521
GetServerTime () method, 743
GetSignature () function, 688
GetSignatureLines () function, 683
GetType () method, 157
GetUpperBound () method, 401
GetWindowsCEApplicationManager () function, 607
GetWindowText () function, 257
Global Assembly Cache (GAC), 471, 492
 cgacutil.exe utility and, 577
 gacutil.exe utility and, 493, 577
 location of, 492
 shared assemblies in, 486, 492–494
global variables, 72
Google Spreadsheets, 560
goto keyword, 69, 101–102, 751
GPRS (General Packet Radio Service)
 connections, 353
greater than (>), 105
greater than or equal to (>=), 105
GridView control, 549–554
group keyword, 754
Grouping operators, 432
GUID, 731, 734
GZipStream class, 353

H

handling events, 229–232
handling exceptions, 377–395. *See also exceptions*
handling multiple exceptions, 381–383
hash value, 344
hashing, 344–347
 encryption v., 347
 salted, 346–347
 for shared assembly, 488
hashing algorithms
 DES, 347
 DSA, 351
 MD5 implementation, 344, 345, 346
 RC2, 347
 Rijndael, 347
 RSA, 351, 352
 SHA1 implementation, 344, 345, 346
 TripleDES, 347
HasMorePages property, 530
heap, 85, 169
Hell, DLL, 486

- HelloWorld project**
 - creation, 62–64
 - csc.exe and, 64–65
 - dissection, 66–67
 - Helper class, 498**
 - helper functions, 121**
 - helper methods, for RSS Reader application, 583–589**
 - HelpFileFormat drop-down listbox, 775**
 - HelpFileFormat property, 779, 780**
 - HelpLink property, 379, 389**
 - hosting WCF services, 700, 713**
 - HResult property, 379**
 - HTC Touch Cruise, 575**
 - HTML Help 2 (Visual Studio .NET Help format), 773**
 - HTML Help format, MSDN-style (.chm), 773, 780**
 - HTML Help Workshop, 774**
 - HTTP protocol, 534, 546, 547, 560, 618, 700, 710, 711, 712**
- I**
- IAddress interface, 174, 176, 177, 178**
 - IAsyncResult, 224**
 - ICircle interface, 205**
 - ICollection<T> interface, 405**
 - IComparable<Employee> interface, 274**
 - IComparable<T> interface, 272, 405, 413–420**
 - IComparer<T> interface, 405, 413–420**
 - IConsoleLoggin interface, 209, 210**
 - ID attribute, 570**
 - IDE. See Visual Studio 2008**
 - identifiers, keywords as, 749–750**
 - IDictionary<T> interface, 405**
 - IDisposable interface, 154, 155**
 - IEnumerable interface, 410, 430, 431**
 - IEnumerable<Contact>, 436**
 - IEnumerable<DataRow>, 426**
 - IEnumerable<string>, 428**
 - IEnumerable<T> collection, 426**
 - IEnumerable<T> interface, 405, 410–413**
 - IEnumerable<XElement>, 426**
 - IEnumerator interface, 410**
 - IEnumerator<T> interface, 405, 410–413**
 - #if, 115–117**
 - if keyword, 69, 751**
 - if-else statement, 92–93**
 - IFileLogging interface, 209, 210**
 - IIS (Internet Information Services), 504, 534, 545, 546, 547, 696, 700, 713**
 - IL (Intermediate Language), 467. See also MSIL**
 - ildasm Base64Codec.dll, 491**
 - ildasm (MSIL Disassembler) tool, 7, 468, 491**
 - IList<T> interface, 405**
 - Image element, 646, 648, 650**
 - ImageBrush element, 634**
 - ImageList1, 508**
 - IManager interface, 175, 176**
 - IMessageService.cs, 722**
 - Immediate window, 53**
 - immutable**
 - anonymous types, 129
 - strings, 250, 257
 - implementation inheritance, 184–204**
 - implementing callbacks**
 - delegates and, 220–222
 - WCF, 728–741
 - implementing classes, 171, 173–175, 181**
 - implementing equals, 15–161**
 - implementing events, 232–235**
 - implicit keyword, 69, 197, 751**
 - implicit type conversion, 89–90**
 - implicit typing, 88–89, 428**
 - IMyStack interface, 275**
 - in keyword, 69, 752**
 - include tag, 768**
 - Increment () method, 304**
 - indexers, 407–410**
 - IndexOf () instance method, 253**
 - IndexOfAny () instance method, 253**
 - inetmgr command, 504**
 - inheritance, 157, 183–211**
 - base class, 184
 - constructors and, 202
 - derived class, 184
 - implementation, 184–204
 - interface, 204–211
 - keywords for, 195
 - process of, 183–184
 - purpose of, 183
 - single-class, 185
 - types of, 184
 - .ini file, 606, 607, 608, 615**
 - initialization**
 - of arrays, 398–399
 - of objects, 153–154
 - Initialization Vector (IV), 348, 349, 350**
 - InitializeComponent () function, 227**
 - initializers, object, 153–154, 173**
 - inline statements, 225**
 - inner exception, 387**
 - InnerException property, 379, 389**
 - InputBox () function, 591**
 - Insert () instance method, 250, 259**
 - Insert () method, 406**
 - inserting strings, into strings, 251**
 - InsertItem () method, 290**
 - InsertNumber () function, 285, 286**
 - InsertOnSubmit () method, 461, 462**
 - Installer Class template, 605**

installers (for RSS Reader)

- custom, 604–608
- MSI, 609–615

instance members, 131–132

instance methods. See *specific instance methods*

InstanceContextMode property, 723, 724

instantiation, 127–128

int (keyword), 69, 752

int (value type), 80

- conversion, 89
- range, 80

integer values, array of, 429

integrated development environment. See Visual Studio 2008

IntelliSense, 37–39, 190, 199, 267, 384, 437, 441, 484

interface inheritance, 204–211

interface keyword, 69, 752

interface members implementation, explicit, 208–211

interfaces, 154, 171–181. See also *specific interfaces*

- abstract classes *v.*, 171, 211
- casting to, 177–178
- collections, 397, 404–405
- as contracts, 171, 181
- defining, 172
- extending, 175–177
- generic, 275–276
- IEnumerable, 430, 431
- IEnumerator, 410
- implementations, 173–175
 - multiple, 174–175
 - overriding, 179–181
- IMyStack, 275
- naming convention, 172
- in System.Collections.Generic namespace, 283

Interlocked class, 304

Intermediate Language (IL), 467. See also MSIL

internal access modifier, 133, 200–201

internal keyword, 69, 133, 200, 201, 752

Internet Explorer, 536, 554, 560, 618, 623

Internet Explorer dialog–Security Warning, 536

Internet Information Services (IIS), 504, 534, 545, 546, 547, 696, 700, 713

into keyword, 69, 754

Intuos Pen Tablet, 675

Invoke button, 704

Invoke () method, 312, 313, 314

InvokeRequired property, 312, 313

I/O operations, 5

- disk, 353, 375
- file, 312, 337, 375
- memory, 375
- network, 375

IPAddress.Loopback, 585

iPaq211 Pocket PC, 575

IPerson interface, 172, 173, 174, 177, 178, 181

IPHostEntry object, 585

IQueryable<T> collection, 426

IRectangle interface, 206, 207

is keyword, 69, 752

is operator, 178–179

IsAlive property, 297

IsConnected () function, 585

IService1.cs, 701, 702, 703, 704, 706

IShape class, 206

IShape interface, 205

IsMatch () method, 260

IsNetworkDeployed property, 540

IsNull () method, 449

isolated storage, of signature, 681–686

ISquare interface, 206

iterators, 407–410

ITicketCallback interface, 731

ITicketService interface, 731, 734

IV (Initialization Vector), 348, 349, 350

J

jagged arrays, 402–403

JavaScript

- Asynchronous, 560. *See also* AJAX
- MySilverlight.js file, 622
- RoundButton and, 644–645
- Silverlight and, 621, 622, 644–645
- Silverlight.js file, 621, 622

JIT (just-in-time) compiler, 4, 9, 467, 499

join keyword, 69, 754

Join () method, 300

Joining operators (LINQ), 432

joining tables, 443–446

just-in-time (JIT) compiler, 4, 9, 467, 499

K

key token, public, 488, 491, 498

KeyCollection class, 421

KeyedCollection<TItem> class, 288

KeyedCollection<TKey> class, 288

KeyNotFoundException error, 421

keys. See also encryption

- private, 347, 350, 351, 487, 488
- public, 350, 351, 352, 487, 488, 491

KeySpline, 650, 653, 654

keywords, 68–69, 749–755. See also *specific keywords*

- access modifiers. *See* access modifiers
- contextual, 749–750, 754–755
- evolution of, 750
- as identifiers, 749–750
- for inheritance, 195
- list, 69, 749–755

for loop exiting, 99–102
 reserved (list), 750–754
 variables as, 70

L

Label control, 312, 313, 314, 320

lambda expressions, 226, 229, 430

anonymous methods and, 226, 229
 method syntax and, 430

lambda operator, 430

Language Integrated Query. See LINQ

last in, first out (LIFO) data structure, 265, 281, 422

LastIndexOf () instance method, 253

Learn2develop.net namespace, 481, 482, 483

Length () method, overloaded, 777, 778

Length property, 399, 403

lengthTest () method, 56, 57

less than (<), 105

less than or equal to (<=), 105

let keyword, 69, 754

libraries. See class libraries

library assembly (DLL), 467

LIFO (last in, first out) data structure, 265, 281, 422

#line, 118–120

<Line> element, 628

#line hidden, 120

<LinearGradientBrush> element, 632

lines array, 402

linked lists, 284

circularly, 284
 doubly, 281, 284
 singly, 284, 285

LinkedList<T> generic class, 284–287

LinkedListNode<T> generic class, 285

LINQ (Language Integrated Query), 425–465

aggregate functions, 435, 442–443

anonymous types and, 436–437

architecture, 425–426

Contact class and, 436–437

extension methods and, 430–436

implicit typing and, 89

operators

Aggregation, 432

Conversion, 432

Element, 432

Equality, 432

Generation, 432

Grouping, 432

Joining, 432

Ordering, 432

Partitioning, 432

Quantifiers, 432

Restriction, 432

Selection, 432

Set, 432

SQL queries v., 428

LINQ to DataSet, 426, 438–450

ADO.NET and, 438

anonymous types and, 441–442

null fields, 449

reshaping data, 441–442

table joining in, 443–446

typed DataSet feature, 446–448

LINQ to Entities, 426

LINQ to Objects, 426–429

LINQ to SQL, 426, 458–465, 548–549

Classes template, 548

database modeling with, 548–549

O/R Designer and, 458–460

LINQ to XML, 426, 450–457

querying elements with, 452–453, 456–457

RSS document (example), 453–457

usefulness of, 453

XML trees, 450–452

linqDataSource control, 547, 550, 555, 556, 558, 572

<list> tag, 768

list traversal, 284, 287

lists, linked, 284

List<T> class, 288, 407

Collection<T> class v., 291

literals, anonymous types and, 130

Load () method, 452, 454

Loaded attribute, 652

local variables, 72

lock keyword, 69, 304–307, 752

lock, mutually exclusive, 305

locked code, 305

LogEmail () method, 143, 144

LogError () method, 208, 209, 210

logical operators, 106–108, 109

&& (AND), 106, 107

! (NOT), 106, 107

|| (OR), 106, 107

short-circuit evaluation, 107–108

long (keyword), 69, 752

long (value type), 80

conversion, 89

range, 80

LongCount () function, 443

loop variant, 258

loops, 96–99

for, 96–98

defined, 96

do-while, 99

exiting from, 99–102

foreach, 98, 400, 433, 437, 441, 442

iterations, skipping, 102

nesting, 97–98

while, 99

M

Main () function, 300, 382

Main () method, 67, 68, 214

managed code, 3

Manager class, 184, 275

manifest

application(PhotoViewer), 541–542

of assembly, 469

deployment, PhotoViewer, 542–543

of WinBase64.exe assembly, 498

manipulating

files/directories, 321–327

strings, 247–254

Match () method, 260

Matches () method, 260

matching, on strings, 259–263. See also regular expressions

mathematical operators, 108–109

MathUtil project, 470–474

MathUtil.dll, 470, 471, 473, 474

MathUtil.netmodule, 476, 479

Max () function, 443

MD5 implementation, 344, 345, 346

Media project (Silverlight 1.0), 654–663

auto-play, disabling, 657–658

creating, 654–656

mirror effect, 658–663

MediaElement element, 655, 658, 663

MediaEnded (), 667, 674

MediaPlayer project (Silverlight 1.0), 664–675

controls, wiring, 667–669

creating, 664

event handlers, 667, 671–675

helper functions, 669–671

slider, 665

user interface, designing, 664–667

XAML code, 666–667

MediaPlayerMouseLeave (), 667, 673

MediaPlayerMouseMove (), 667, 672–673

Member class, 367

MemberAddress class, 367

MemberName class, 367

members (class), 131–132

data, 131

function, 131, 134–155

instance, 131–132

signatures, 148, 205

static, 132

memory

allocation, 85, 169

I/O, 375

MemoryStream class, 331, 337–338

Menu bar, 17–19

MenuItem control, RSS Reader, 581, 582

message contract, 712

Message property, 379

MessageServer project, 720

MessageService.cs, 721, 723

messaging patterns, 713. See also communication models

Method (), 201

method extraction feature, 40–42

Method () method, 193, 194

method syntax, 429–430

MethodDelegate, 215, 216

methods, 134–138. See also specific methods

abstract, 186–189

anonymous, 225–226

definition of, 134

extension, 198–199

functions v., 134

generic, 276–278

overloaded, 148, 163, 190, 192–195, 777, 778

partial, 143–145

passing arguments into, 68, 135–137

sealed, 191–192

virtual, 145, 189–190

web, 700

methods delegate, 218, 219

MexHttpBinding, 712, 717

Microsoft Intermediate Language. See MSIL

Microsoft Silverlight. See Silverlight

Microsoft Visual Studio 2008. See Visual Studio 2008

Microsoft.VisualBasic.dll library, 591

Min () function, 443

mirror effect, 658–663

missing controls, 25

mobile application platforms, 573. See also Windows Mobile

ModalPopupExtender control, 566–572

modeling databases, 548–549

modules, assemblies and, 468, 474–480

modulus operator (%), 108, 109

Monitor class, 307–311

Mono (.NET Framework implementation), 4

Moto Q9h, 576

MouseLeftButtonDown event, 677, 678, 679

MouseLeftButtonDown event handler, 684, 685, 686

MouseLeftButtonUp event, 677, 678, 680

MouseMove event, 677, 678, 680

Move () method, 327

MoveNext () method, 410, 412

MSDN–Online style Web pages, 773

MSDN–style HTML Help format (.chm), 773, 780

MSI installer, 609–615

MSIL (Microsoft Intermediate Language), 4

assemblies and, 6–8, 468

Disassembler tool, 7, 468, 491

generics and, 265. *See also* generics

MSMQ, 710, 712
MsmqIntegrationBinding, 712
multicast delegates, 217–220
multidimensional arrays, 400–401
multi-file assemblies, 469–480
multiple endpoints, 714–720. See also endpoints
multiple exceptions, handling, 381–383
Multiple Startup Projects option, 27
Multiple thread value, 734
MultipleEndpointsService, 714, 716, 717
multiplication operator (*), 108
multitasking, 293
multithreading, 293–294, 295, 303, 313. See also
threading
multi-user server application, 340–343
mutually exclusive lock, 305
myDelegate, 314
MyDictionary class, 275
MySilverlight.js file, 622
MyStack class, 266, 268, 269

N

named pipes, 710, 711, 712
namespace keyword, 66, 69, 752
namespaces, 5, 66. See also specific namespaces
 aliases, 485
 assemblies and, 480–486
 Object Browser and, 761, 762
naming convention, interface, 172
narrowing, 90
nested for loop, 97–98
.NET CF. See .NET Compact Framework
.NET Compact Framework (.NET CF), 576–577
 versions, 576–577
 Windows Mobile 6 Standard application and, 581
.NET Framework
 (1.0), 8, 758
 (1.1), 8, 758
 (2.0), 8, 758
 (3.0), 8, 546, 758–759
 introduction of, 546
 printing in, 526–527
 versions, 8–9, 757–759
.NET Framework (3.5), 3–9
 ADO.NET. *See* ADO.NET
 ASP.NET. *See* ASP.NET
 assemblies. *See* assemblies
 class libraries, 4–5, 66
 BCL, 5, 758, 759
 FCL, 5
 generics and, 280–284
 Object Browser and, 759–764
 components, 5, 9

download, 547
 features, 759
 LINQ. *See* LINQ
 Mono implementation, 4
 value types, 80–81
 VB.NET. *See* VB.NET
 versions, earlier, 8–9, 757–759
.NET Help format (HTML Help 2), 773
NetMsmqBinding, 712
NetNamedPipeBinding, 712
NetPeerTcpBinding, 712
NetTcpBinding, 712, 714, 724, 725, 735
network I/O, 375
NetworkCredential class, 518
NetworkStream class, 331, 338–343
 socket applications
 client-server, 338–340
 multi-user server, 340–343
new keyword, 69, 87, 127, 153, 169, 180, 190, 194,
398, 400, 401, 752
new project creation, 14–16
New Project dialog, 15, 16, 635, 676
New Service Endpoint, 715
NORMAL symbol, 116, 117, 118
Northwind database, 439, 443, 444, 447
not equal (!=), 105
NOT operator (!), 106, 107
Notepad, 64, 621, 622, 624
NotSupportedException, 338
null coalescing operator (??), 83
null fields, detecting, 449
null keyword, 69, 84, 752
nullable value type, 81–84
num array, 398, 399, 404
numbers
 formatting, 255, 256, 257
 random generator, 285, 286, 287, 343, 346,
 348, 349
numeric data types. See value types

O

Object Browser, 759–764
 Add to Reference button, 764
 assemblies and, 763–764
 Edit Custom Component Set dialog, 761
 information in, 760
 namespaces and, 761, 762
 search capabilities, 762
objects, 125. See also specific objects
 creating, from classes, 127–128
 destroying, CLR and, 154
 exception, 389–391
 flattening. *See* serialization

objects (continued)

- initialization of, 153–154
 - sorting, 413–420
 - object initializers, 153–154, 173**
 - object instantiation, 127–128**
 - object keyword, 69, 752**
 - Object Relational Designer (O/R Designer), 458–460, 549**
 - object type, 85, 88, 89**
 - ObjectDataSource, 550**
 - object-oriented programming, 67**
 - classes in, 125. *See also* classes
 - delegates in, 213. *See also* delegates
 - encapsulation in, 139
 - events in, 213. *See also* events
 - inheritance in, 183, 184. *See also* inheritance
 - interfaces in, 171. *See also* interfaces
 - objects in, 125. *See also* objects
 - Objects and Timeline window, 636, 640, 642, 647**
 - ObservableCollection<T> class, 288**
 - Obsolete attribute, 163, 164**
 - OkControlID attribute, 570**
 - OnCancelScript attribute, 571**
 - one-way communication model, 700, 713, 723, 731**
 - OnOkScript attribute, 571**
 - onTimeUp () method, 234**
 - Opacity attribute, 661**
 - OpenFileDialog class, 524**
 - OpenNetCF's Smart Device Extension, 577**
 - OpenRead () method, 332**
 - OpenWrite () method, 332**
 - operation contract, 712**
 - OperationContract attribute, 703, 707**
 - operations (web methods), 700**
 - operator keyword, 69, 197, 752**
 - operators, 102–109. *See also* specific operators**
 - assignment, 103–105, 109
 - colon (:), 185
 - conditional, 93, 109
 - generic, 278–279
 - lambda, 430
 - LINQ
 - Aggregation, 432
 - Conversion, 432
 - Element, 432
 - Equality, 432
 - Generation, 432
 - Grouping, 432
 - Joining, 432
 - Ordering, 432
 - Partitioning, 432
 - Quantifiers, 432
 - Restriction, 432
 - Selection, 432
 - Set, 432
 - LINQ (list), 432
 - logical, 106–108, 109
 - mathematical, 108–109
 - overloading, 195–198
 - postfix, 104–105
 - precedence of, 109–110
 - prefix, 104–105
 - primary, 109
 - regex (list), 261
 - relational, 105–106, 109
 - self-assignment, 103–104
 - unary, 109
 - O/R Designer (Object Relational Designer), 458–460, 549**
 - OR operator (| |), 106, 107**
 - Order Details table, 444, 445, 446**
 - orderby keyword, 69, 754**
 - Ordering operators, 432**
 - Orders table, 444, 445, 446**
 - Organize Usings, 38**
 - otherContact parameter, 152**
 - out keyword, 69, 137, 752**
 - Output window, 30**
 - OutputPath property, 779**
 - overloaded methods, 148, 163, 190, 192–195**
 - Length (), 777, 778
 - overloading operators, 195–198**
 - overloads· element, 771, 778**
 - override keyword, 69, 195, 752**
 - overriding interface implementations, 179–181**
- ## P
- PadLeft () instance method, 251**
 - PadRight () instance method, 251**
 - Page () constructor, 683**
 - Page.xaml.js file, 644, 649, 652, 657, 658, 662, 669**
 - painting shapes, 630–634**
 - ImageBrush· element, 634
 - LinearGradientBrush· element, 632
 - RadialGradientBrush· element, 633
 - SolidColorBrush· element, 631
 - para· tag, 768**
 - param· tag, 768**
 - parameter arrays, 403–404**
 - ParameterizedThreadStart delegate, 301, 302, 303**
 - parameters. *See also* specific parameters**
 - arguments v., 135
 - reordering/removing, 42–44
 - type, in generic type, 275
 - variable promotion to, 48
 - paramref· tag, 768**
 - params keyword, 69, 403, 752**
 - params parameter, 404**

- parent class, 188**
 - Parent property, 322**
 - parent scope, 74**
 - parentheses, forcing precedence with, 110**
 - Parse () method, 92, 457**
 - partial classes, 126–127**
 - partial keyword, 69, 126, 145, 755**
 - partial methods, 143–145**
 - declaring, 144
 - rules for, 145
 - Partitioning operators, 432**
 - passing arguments into methods, 135–137**
 - Main () method, 68
 - by reference, 136
 - by value, 136
 - passing parameters to threads, 301–303**
 - Password (application setting), 509**
 - PCs**
 - Pocket, 574–575. *See also* Smartphones
 - Pocket PC 2000, 574
 - Pocket PC 2002, 574
 - Windows Mobile 5.0 for, 574
 - Windows Mobile 6 Classic, 574, 575
 - Windows Mobile 6 Professional, 574, 575
 - Windows Mobile 2003 for, 574
 - Windows Mobile 2003 SE for, 574
 - tablet, 675, 677. *See also* Signature project
 - PE format. *See* Portable Executable format**
 - Peek () method, 423**
 - Pen Tablet, Wacom, 675**
 - Percall service instance, 724**
 - PerformDivision () function, 382, 383, 386**
 - PerformMathOps () function, 216**
 - PerformWebRequest () function, 517, 518, 521, 522, 523, 525**
 - Perimeter () method, 189, 190, 207**
 - Perl, 546**
 - permission· tag, 768**
 - PerSession service instance, 724**
 - PhotoViewer (Windows application), 503–504**
 - application manifest, 541–542
 - application settings and, 508–513
 - ClickOnce and, 503, 534–536
 - coding of, 513–514
 - completed project, 504, 526
 - controls for, 506–508
 - creation of, 506–508
 - deploying, 533–543
 - deployment location, 536
 - deployment manifest, 542–543
 - directory
 - creating, 522
 - removing, 523
 - directory tree for, 515–521
 - displaying images in, 515–521
 - FTP Server
 - configuration of, 504–506
 - deleting photos, 525–526
 - information for access, 508–509
 - uploading photos to, 524–525
 - print support for, 526–533
 - programmatically updating, 538–540
 - publishing, 534–536
 - rolling back, 541
 - source code for, 507
 - testing, 526
 - updating, 536–538
 - user.config file, 512, 513
- PhotoViewer.exe.config, 512**
- PhotoViewer.exe.deploy file, 543**
- Phrases () method, 408**
- PictureBox control, 521**
- PictureBox1, 507, 508**
- P/Invoke (Platform Invoke), 577**
- pipes, named, 710, 711, 712**
- Plain Old XML (POX), 618, 700**
- Platform Invoke (P/Invoke), 577**
- PlayPauseButtonUp (), 667, 674**
- plug-in, Silverlight. *See* Silverlight plus operator. *See also* addition operator**
- Pocket PCs. *See also* Smartphones**
 - Pocket PC 2000, 574
 - Pocket PC 2002, 574
 - Windows Mobile 5.0 for, 574
 - Windows Mobile 6 Classic, 574, 575
 - Windows Mobile 6 Professional, 574, 575
 - Windows Mobile 2003 for, 574
 - Windows Mobile 2003 SE for, 574
- Point array, 402**
- Point class, 195, 196, 197, 402**
 - definition
 - members, 767
 - XML comments added, 767–771
- PointClass project, 765**
- pointers, to functions. *See* delegates**
- Polygon· element, 629–630**
- Polyline· element, 630**
- Pop () method, 263, 423**
- PopupControlID attribute, 570**
- Portable Executable (PE) format, 467**
- portable media centers, 574**
- postfix operators, 104–105**
- POX (Plain Old XML), 618, 700**
- #pragma warning, 122–123**
- precedence**
 - forcing, 110
 - operator, 109–110
- predefined value types (list), 80–81**
- prefix operators, 104–105**

preprocessor directives, 110–123. See also specific preprocessor directives
defined, 110
list, 110

Presentation Core, 620

Presentation File Patches, 774

primary operators, 109

Print () method, 527

PrintDialog class, 532

PrintDocument class, 503, 526, 527, 529

PrintFoldersinCurrentDirectory () function, 330

printing, in .NET, 526–527

PrintMessage () function, 404

PrintName () method, 135, 163

PrintPage event, 527

PrintPageEventArgs class, 529, 530

PrintPreviewDialog () class, 531

PrintString () method, 189

PrintTime () function, 313, 314

private access modifier, 133, 142, 146

private assemblies, 486. See also assemblies

private key, 347, 350, 351, 487, 488

private key encryption, 347, 487. See also symmetric encryption

private keyword, 69, 133, 142, 146, 200, 201, 752

process assemblies, 467

Process.Start () method, 608

Programmer attribute, 164, 165

ProgressChanged event, 318, 319

ProgressTemplate element, 566

projects
creation, 14–16
startup, 27

Promote Local Variable to Parameter, 48

properties, 139–146. See also specific properties
automatic, 145–146, 173
read-only, 142–143
write-only, 142–143

Properties Inspector, 637, 640, 648

Properties window, 17, 28–29

Properties.Settings.Default class, 514

protected access modifiers, 133, 200–201

protected internal keyword, 200, 201

protected keyword, 69, 133, 200, 201, 752

protocols. See communication protocols

·pubDate· element, 457

public access modifier, 133

public key, 350, 351, 352, 487, 488, 491

public key encryption, 350, 487. See also asymmetric encryption

public key token, 488, 491, 498

public keyword, 69, 133, 201, 752

Publish Wizard, 534, 537, 541

publisher, 226

publisher table, 549, 554

publishers
field, editable, 559–560
names, displaying, 554–555
titles from, displaying, 555–558

publishers table, 458, 459

Publish.htm page, 536

publishing PhotoViewer, 534–536

pubs sample database, 438, 439, 458, 548, 549, 572

Pulse () method, 310

Push () method, 423

Q

Quantifiers (LINQ operators), 432

queries. See also LINQ; SQL
deferred execution, 432–434
forced immediate execution, 434–436
LINQ to SQL, 460–461
saved, to DataTable, 449–450
SQL v. LINQ, 428
syntax, 429–430
for Wrox books, 452–453

querying elements, with LINQ to XML, 452–453, 456–457

Queue ·T· class, 283, 284, 423, 424

queues, 423–424

QuickWatch feature, 52

R

·RadialGradientBrush· element, 633

RadiusX attribute, 627

RadiusY attribute, 627

RaiseItemRemovedEvent () function, 291

random-number generation, 285, 286, 287, 343, 346, 348, 349

ranges, of value types, 80

Rank property, 399

rank specifier, 398

rate of fall, 653–654

RC2 algorithm, 347

Read () method, 328, 332, 339

ReadAllBytes () method, 328

ReadAllLines () method, 328

ReadAllText () method, 328

ReadBlock () method, 328

ReadByte () method, 332

ReadFile () method, 156

reading/writing to files, 327–328

ReadLine () method, 67, 328

readonly keyword, 69, 753

read-only properties, 142–143

ReadOnlyCollection·T· class, 288

ReadOnlyObservableCollection·T· class, 288

- ReadToEnd () method, 328**
- Rearrange Commands dialog**
 - Menu bar, 18–19
 - Toolbar, 21
- Record Keyframe button, 647, 648**
- Rectangle class, 185, 186, 188, 192, 198, 204**
- Rectangle element, 624, 625, 627–628**
- Reentrant thread value, 734**
- ref keyword, 69, 136, 137, 753**
- refactoring support, 39–48**
- reference equality, 157, 159**
- reference types, 84–85. See also arrays**
 - predefined, 85
 - value types v., 85
- ReferenceEquals () method, 158, 159**
- Reflection folder, 774**
- Refresh Feed MenuItem control, 595**
- RefreshCurrentFolder () function, 522**
- refreshing web pages, 560**
- regex. See regular expressions**
- Regex class, 259, 260**
- #region, 120–122**
- RegisterClient operation, 731, 734**
- regular expressions (regex), 259–263**
 - defined, 241
 - email validation with, 263
 - operators (list), 261
 - search patterns (list), 262
- relational operators, 105–106, 109**
- remarks tag, 768**
- Remove Folder button, 523**
- Remove () instance method, 254**
- Remove () method, 259, 406**
- RemoveAt () method, 406**
- RemoveItem () method, 290**
- RemoveSpecialChars () function, 587**
- removing parameters, 42–44**
- renaming variables, 39–40**
- reordering parameters, 42–44**
- repetitive code, 40. See also Extract Method**
- Replace () method, 259**
- replacing strings, 254**
- ReportProgress () method, 318**
- request/response communication model, 700, 713, 728**
- reserved keywords (list), 750–754**
- Reset () method, 410, 412**
- Reset Toolbox, 25**
- reshaping data, 441–442**
- resources, 467**
- Restart () method, 540**
- Restriction operators, 432**
- ResultCallback () function, 221, 222, 224**
- rethrowing exceptions, 386–387**
- RetrieveBytesFromStream () function, 355, 356**
- return keyword, 69, 100, 753**
- returns tag, 768**
- RIAs (Rich Internet Applications), 617, 694**
- Rich Internet Applications (RIAs), 617, 694.**
 - See also Silverlight
- Rijndael algorithm, 347**
- RijndaelManaged class, 348, 349, 350**
- rolling back PhotoViewer, 541**
- Rotate tab, 651**
- RoundButton project (Silverlight 1.0), 635–645**
 - Brush Transform tool, 639
 - Canvas control, 644
 - Ellipse object, 636, 637, 638, 639, 640
 - Expression Blend 2 and, 635–644
 - Page.xaml.js file, 644
 - TextBlock object, 641, 642
 - user interface, JavaScript and, 644–645
- rows (LINQ to SQL)**
 - deleting, 463–465
 - insertion of, 461–462
 - updating, 463
- RSA, 351, 352**
- RSA algorithm, 351, 352**
- RSACryptoServiceProvider class, 352**
- RSS document (LINQ to XML), 453–457**
- rss element, 456**
- RSS Reader application, 580–616**
 - ActiveSync and, 596, 598, 603, 604, 606, 612, 614
 - capabilities of, 580
 - controls for, 581–582
 - deployment, 598
 - CAB file, 598–603
 - target device prerequisites, 615–616
 - event handlers, wiring, 589–596
 - helper methods for, 583–589
 - MenuItem control, 581, 582
 - setup application, 603–616
 - custom installer, 604–608
 - MSI installer, 609–615
 - testing
 - with devices, 597–598
 - with emulators, 596–597
 - TreeView control, 581, 582, 587, 589, 591, 592
 - URL of, 587
 - user interface, 581–583
 - WebBrowser control, 582
- RSSReaderInstall.dll file, 611**
- RSSReaderInstaller project, 604–608**
- RSSReaderSetup project, 609, 610, 611, 612**
- RSSReaderSetup.msi application, 612, 614**
- Run All Tests in Solution button, 57**
- RunWorkerAsync () method, 317**
- RunWorkerCompleted event, 319**

S

Safari browser, 618

safety

- thread, 303, 312–314
- type, 271, 410

SalarySorter property, 419

salted hash, 346–347

Salted_Hashing_SHA1 () function, 346–347

Samsung Black II, 597

Sandcastle Help File Builder, 774, 775

Sandcastle Styles Project site, 774

Sandcastle tool, 78, 765, 773

- downloading, 774
- installing, 774
- launching, 775–776

Save () method, 337, 451

SaveSignature () function, 688

sbyte (keyword), 69, 753

sbyte (value type), 80

- conversion, 89
- range, 80

scope, 71

- for application settings, 510
- parent/current, 74
- of variables, 71–74

scrambling, 347. See also encryption

scripting languages, 545, 546

ScriptManager control, 563

SDKs (Software Development Kits)

- Silverlight 1.0, 619, 621
- Silverlight 2.0 Beta 1, 620
- Windows Mobile 5.0 for Pocket PC, 578
- Windows Mobile 5.0 for Smartphone, 578
- Windows Mobile 6 Professional, 578
- Windows Mobile 6 Standard, 578, 581

sealed classes, 191–192

sealed keyword, 69, 191, 192, 195, 753

sealed methods, 191–192

searching

- with Object Browser, 762
- with regex, 259–263
- for strings, 252–254

Security Accounts tab, 505

security services, 343. See also cryptography services

«see» tag, 768

«seealso» tag, 768

Seek () method, 332, 338

Select attribute, 556

select keyword, 69, 755

select-from-where format, 428

Selection operators, 432

self-assignment operators, 103–104

self-hosted WCF services, 713, 720–728

serialization, 359–375

- binary, 359–365
- deep, 364
- defined, 359
- deserialization and, 370–371
- shallow, 364
- XML, 359, 365–375

Serialize () function, 360, 361

service contract, 712

service oriented applications (SOA), 695.

See also WCF

Service Reference Settings dialog, 718

Service1.cs, 701, 703

ServiceBehavior attribute, 723, 734

ServiceContract attribute, 703, 731

ServiceHost class, 721

ServiceReference1, 709

Service.Runtime.Serialization.dll, 709

Service.ServiceModel.dll, 709

SessionMode property, 731

set accessor, 140, 141, 142, 146, 749

Set as Startup Project, 27

set keyword, 69, 749, 750, 755

Set operators, 432

SetCurrentDirectory () method, 324, 325

setInterval () function, 674

SetItem () method, 290

SetLength () method, 338

SetMessage () operation, 723

SetSeatStatus (), 731, 734, 738, 739

setting breakpoints, 49–50

Settings.Designer.cs file, 510

Settings.settings file, 510

setup application (for RSS Reader), 603–616

- custom installer, 604–608
- MSI installer, 609–615

setup.ini file, 608

SHA1 implementation, 344, 345, 346

Shadows keyword, 194

shallow serialization, 364

Shape class, 185, 186, 204

shape elements, painting, 630–634

- «ImageBrush», 634
- «LinearGradientBrush», 632
- «RadialGradientBrush», 633
- «SolidColorBrush», 631

shape elements (Silverlight), 627–630

- «Ellipse», 624, 625, 629
- «Line», 628
- «Polygon», 629–630
- «Polyline», 630
- «Rectangle», 624, 625, 627–628

Shareable service instance, 724**shared assemblies, 486–492. See also Global****Assembly Cache**

- building, 491–492
- creating, 488–492
- deploying, 488
- in GAC, 486, 492–494
- hash for, 488
- strong name for, 486, 489–490
- visibility of, Visual Studio and, 495

short (keyword), 69, 753**short (value type), 80**

- conversion, 89
- range, 80

short-circuiting, 107–108**Show All Files button, 510****ShowMissingNamespaces property, 776****Signature project (Silverlight 2.0), 675–693**

- ClientBin folder, 692
- coding for, 678–681
- creation, Visual Studio and, 676–693
- hosting, 676
- service reference, 689
- signature
 - capture of, 677–678
 - isolated storage of, 681–686
 - sending to Web Services, 687–693

signatures, 148, 194, 205**SignatureWebSite project, 687, 691, 692, 693****Silverlight (1.0), 618–619, 645–675**

- Animations project, 645–650
- Animations2 project, 650–654
- architecture, 620–621
- bare-bones application
 - Canvas control, 624–626
 - creating, 621–623
 - ⟨Ellipse⟩, 624, 625, 629
 - Expression Blend 2 and, 635–644
 - files in, 621
 - ⟨Line⟩, 628
 - ⟨Polygon⟩, 629–630
 - ⟨Polyline⟩, 630
 - ⟨Rectangle⟩, 624, 625, 627–628
 - shape elements, 627–634
- features (list), 618–619
- for Mac, 619
- Media project, 654–663
- MediaPlayer project, 664–675
- SDK, 619, 621
- user interface
 - JavaScript and, 621, 622, 644–645
 - XAML and, 621, 622, 623–624
- version 2.0 v., 617–619, 694
- for Windows, 619

Silverlight (2.0 Beta 1), 617–694

- AJAX v., 675
- architecture, 620–621
- Beta 1 runtime, 620
- browsers and, 618
- Expression Blend 2 and, 620, 635–644
- Expression Design and, 620
- Expression Media Encoder Preview Update and, 620
- features (list), 618–619
- interactive capabilities of, 675
- for Mac, 619
- RIAs through, 617, 694
- SDK Beta 1, 620
- signature captures and, 675
- Signature project, 675–693
- Tools Beta 1 for Visual Studio 2008, 620, 675, 676
- tools for, 619–620
- version 1.0 v., 617–619, 694
- for Windows, 619
- WPF/E and, 617

Silverlight.js file, 621, 622**Simple Object Access Protocol (SOAP), 618, 699, 700, 706, 712****simplex communication model, 700, 713, 723, 731****Single () method, 463****Single service instance, 724****Single thread value, 734****single-class inheritance, 185****single-file assemblies, 469****singly linked lists, 284, 285****size of keyword, 69, 753****skipping an iteration, 102****Sleep () method, 224, 297, 337, 566****slider (MediaPlayer), 665****smart clients, 534****Smart Device CAB Project template, 599****Smart Device Extension, OpenNetCF's, 577****Smart Device Project template, 578****SmartDeviceCab1 project, 599, 601, 602, 608, 610****Smartphones. See also Pocket PCs; RSS Reader application**

- Smartphone 2002, 574
- Windows Mobile 5.0 for Smartphone, 574
- Windows Mobile 6 Standard, 574, 576
- Windows Mobile 2003 for Smartphone, 574
- Windows Mobile 2003 SE for Smartphone, 574

SmartTag, 549, 552, 553, 556**sn -k KeyFile.snk, 490****sn -T Base64codec.dll, 491****SNK file, 490****SOA (service oriented applications), 695. See also WCF****SOAP (Simple Object Access Protocol), 618, 699, 700, 706, 712**

socket applications

- client-server, 338–340
- multi-user server, 340–343

Software Development Kits. *See* SDKs

·SolidColorBrush· element, 631

Solution Explorer, 17, 26–28

solutions, 16, 26

Sort () method, 399, 413, 415

sorting

- arrays, 399–400
- objects, 413–420

Source property, 379

SpamPhraseList class, 407–408, 409, 410, 411, 412

Split () instance method, 252

splitting strings, 252

SQL (Structured Language Query), 425. *See also* LINQ to SQL

- LINQ queries v., 428

SQL Server 2005 Express, 439

SqlDataSource, 550

Square class, 191, 192, 204, 207

Stack ·T· class, 283, 284, 422

stackalloc keyword, 69, 753

stacks, 422–423

- custom, 265–266
- defined, 422
- heap and, 85, 169
- as LIFO data structures, 265, 281, 422
- memory and, 85, 169, 753

StackTrace property, 379

Start () method, 295

starting threads, 295–296

StartsWith () instance method, 253

startup projects, 27

state information, passed to event handler, 237–240

state management, 546

state, of object, 375

stateless nature, of HTTP protocol, 534, 546, 700

static classes, 156–157

static constructors, 151–152

static keyword, 67, 69, 131, 132, 195

static members, 132

Step Into, 51

Step Out, 51

Step Over, 51

StocksPriceChecker project, 697, 699, 700

StockWebService project, 695–697, 699, 700

_stopThread variable, 299, 300, 301

Stopwatch object, 258

Storyboard1, 647, 649, 651, 653

Storyboard2, 649

strArray, 252

Stream class, 351–353

StreamReader class, 328–329

StreamReader object, 587

streams, 331–358

- BufferedStream class, 331, 333–334

- compressions, 353–354

- CryptoStream class, 331, 348–349

- decompressions, 354–358

- defined, 321, 331

- files v., 321

- FileStream class, 331, 334–337

- MemoryStream class, 331, 337–338

- NetworkStream class, 331, 338–343

- types of, 321

StreamWriter class, 328–329

string arrays, 68, 427, 428, 429, 430

String class, 247

- methods, 247

string keyword, 69, 753

StringBuilder class, 257–259

String.Compare () static method, 249

String.Join () static method, 251

strings, 85, 241–259

- as C# type, 85

- case changing of, 254

- comparing, 248–249

- concatenating, 250–251, 257

- conversion of, to value type, 92

- creating, 250–251

- defined, 241

- equality, testing for, 248

- formatting, 254–257

- immutable, 250, 257

- inserting strings into, 251

- manipulations, 247–254

- matching on, 259–263

- regex and, 259–263

- replacing, 254

- searching for, 252–254

- splitting, 252

- trimming, 251

- as value types, 242

stringToByteArray () function, 357

StringUtil project, 475

StringUtil.netmodule file, 475, 479

strong name (shared assembly), 486, 489–490

strong typing, 78, 89, 280, 281, 447

struct keyword, 69, 753

structs (structures), 79, 165–169. *See also specific structs*

- classes v., 165–166, 169

- Coordinate, 166, 167, 168, 276

- default constructor and, 166

- generic, 276

- memory allocation and, 169

- in System.Collections.Generic namespace, 282

Structured Language Query. *See* SQL

structures. *See* structs

SubmitChanges () method, 462
subroutines, 67. See also functions
SubscribeFeed () function, 587
Subtract () method, 214, 216
subtraction operator (-), 108
Sum () function, 443
 ‹summary› tag, **767, 768**
SumNumbers () function, 317, 319
Surround With code snippets feature, 36–37
switch keyword, 69, 94, 753
switch statement, 94–95
symmetric encryption, 347–350, 487
SymmetricDecryption () function, 350
SymmetricEncryption () function, 348, 349, 350
synchronization, thread, 303–312
 Interlocked class and, 304
 lock keyword and, 304–307
 Monitor class and, 307–311
synchronous callbacks, 222
syntax
 C# language, 68–123
 method, 429–430
 query, 429–430
System.Attribute, 164
System.Collections namespace, 404, 405, 424
 System.Collections.Generic namespace v.,
 283–284
System.Collections.Generic namespace
 classes in, 280–281
 interfaces in, 283
 structs in, 282
 System.Collection namespace v., 283–284
System.Collections.ObjectModel namespace, 288–291
System.Configuration.Install, 605
System.Convert class, 91
System.Core.dll, 431
System.Data.DataSetExtensions.dll, 440
System.Drawing.Printing namespace, 503, 528
System.IO namespace, 230, 331, 497
System.IO.Compression namespace, 353
System.Object class, 157–158
 instance methods, 157
 static methods, 158
System.Object type, 85
**System.Runtime.Serialization.Formatter.Binary
 namespace, 360, 361**
System.String class, 241–243
System.Text.RegularExpressions namespace, 259
System.Threading namespace, 295
System.Timers namespace, 229
System.Windows.Forms, 605
System.Xml namespace, 370
System.Xml.Linq namespace, 450
System.Xml.Linq.dll, 450
System.Xml.Serialization namespace, 367

T

table(s)
 authors, 438, 439, 440, 458, 459
 Customers, 444, 445, 446, 449
 joining, 443–446
 Order Details, 444, 445, 446
 Orders, 444, 445, 446
 publishers, 458, 459
 rows
 deleting, 463–465
 insertion of, 461–462
 updating, 463
 titleauthor, 458, 459
 titles, 458, 461, 551, 553, 554, 555, 572
tablet PCs, 675, 677. See also Signature project
TargetControlID attribute, 570
TargetSite property, 379
TCHAR* parameter, 257
TCP protocol, 331, 338, 339, 710, 711, 726
TcpClient class, 339, 342
TcpListener class, 339, 342
templates, classes as, 125, 127
ternary operator, 93
Test button, 497, 498
test methods, 56
Test Tools toolbar, 57
TestDefine program, 113–114
testing. See also unit tests
 for equality, 158–159
 strings, 248
 PhotoViewer, 526
 RSS Reader application
 with devices, 597–598
 with emulators, 596–597
 WCF ticketing service, 740–741
TextBlock object, 641, 642
TextWriter class, 329
**this keyword, 69, 137–138, 150, 187, 199, 408,
 749, 753**
Thread class, 293, 295
thread safety, in Windows Forms, 303, 312–314
ThreadAbortException exception, 298
threading, 293–320
 asynchronous callbacks and, 225
 multi, 293–294, 295, 303, 313
threads
 aborting, 297–301
 defined, 293
 passing parameters to, 301–303
 starting, 295–296
 synchronization, 303–312
 Interlocked class and, 304
 lock keyword and, 304–307
 Monitor class and, 307–311

- ThreadStart delegate, 295, 301, 302
 - throw keyword, 69, 100–101, 753
 - throwing exceptions, 383–387
 - ticketing service. *See* WCF ticketing service
 - tilde, 154
 - Timeline object, 645, 648, 649, 651, 653
 - Timer class, 229, 230, 232
 - TimesUp event, 235, 238, 240
 - titleauthor, 458, 462, 572
 - titleauthor table, 458, 459
 - titles table, 458, 461, 551, 553, 554, 555, 572
 - ToArray () method, 406
 - token, public key, 488, 491, 498
 - ToLower () instance method, 254
 - ToolBar, 17, 19–22
 - Toolbox, 17, 22–25
 - tools
 - Brush Transform, 639
 - debugging. *See* Visual Studio 2008
 - ildasm (MSIL Disassembler), 7, 468, 491
 - Sandcastle, 78, 765, 773–776
 - for Silverlight, 619–620
 - Tools Beta 1 for Visual Studio 2008, 620, 675, 676
 - ToolStripStatusLabel1, 507
 - ToString () method, 157, 161–162, 259, 380, 387, 451
 - touch screens, 574, 575, 580
 - ToUpper () instance method, 254
 - ToXmlString () method, 352
 - Translate tab, 651
 - transport protocols. *See* communication protocols
 - TreeView control, RSS Reader, 581, 582, 587, 589, 591, 592
 - TreeView1, 507, 508
 - TreeView1_AfterSelect event, 520
 - TreeView1_BeforeExpand event, 516
 - triggers, 564–565
 - Trim () instance method, 251
 - TrimEnd () instance method, 251
 - trimming strings, 251
 - TrimStart () instance method, 251
 - TripleDES algorithm, 347
 - true keyword, 69, 753
 - truth tables, 106, 107
 - try keyword, 69, 753
 - try–catch block, 36, 100, 101, 219, 378–381, 383, 391, 751
 - TryGetValue () method, 421
 - TryParse () method, 92
 - two–way communication model, 700, 713
 - txtFrom control, 527
 - txtFTPServer, 507
 - txtFtpServer_Leave event handler, 514
 - txtNewFolderName, 507
 - txtPassword, 507, 508
 - txtTo control, 527
 - txtUserName, 507
 - types. *See* data types; *specific types*
 - type casting, 90. *See also* casting
 - type conversion, 89–92
 - explicit, 90–92
 - implicit, 89–90
 - type metadata, 467, 468
 - type parameters, in generic type, 275
 - type safety, generics and, 271, 410
 - typed DataSet feature, 446–448
 - TypedCustomersDataSet.xsd, 447, 448
 - typeof keyword, 69, 753
 - ‹typeparam› tag, 768
 - ‹typeparamrefs› tag, 768
 - typing
 - explicit, 88
 - implicit, 88–89, 428
 - strong, 78, 89, 280, 281, 447
- ## U
- UI controls, updating, 312, 314, 317, 318, 320
 - uint (keyword), 69, 753
 - uint (value type), 80
 - conversion, 89
 - range, 80
 - ulong (keyword), 69, 753
 - ulong (value type), 80
 - conversion, 89
 - range, 80
 - unary operators, 109
 - unchecked keyword, 69, 91, 753
 - #undef, 111–115
 - underscore (_), 133
 - unflattening data, 362
 - Unicode, 242, 247. *See also* strings
 - Uninstall event, 606
 - unit tests, 53–59
 - creating, 54–56
 - with floating point numbers, 58–59
 - running, 58
 - unmanaged code, 3, 4, 154, 155
 - UnRegisterClient operation, 731, 734
 - unsafe keyword, 69, 754
 - Unsubscribe MenuItem control, 594
 - Update () method, 540
 - updateLabel, 314
 - UpdatePanel control, 563–564
 - UpdateProgress control, 565–566
 - updating UI controls, 312, 314, 317, 318, 320
 - UploadFile () method, 524
 - UploadImage () function, 524
 - user interface
 - MediaPlayer project, 664–667
 - RoundButton project, 644–645

RSS Reader application, 581–583
 Silverlight 1.0
 JavaScript and, 621, 622, 644–645
 XAML and, 621, 622, 623–624

user.config file (PhotoViewer), 512, 513

user-defined data types, 78

UserName, 509

ushort (keyword), 69, 754

ushort (value type), 80

 conversion, 89
 range, 80

using keyword, 66, 69, 155, 754

Utils.dll, 476, 478, 479, 480

V

ValidateEmail () method, 134

value equality, 159

value keyword, 69, 141, 749, 750, 755

value tag, 768

value types, 79–84. See also specific value types

 default value of, 81–82
 fixed size of, 242
 .NET, 80–81
 nullable, 81–84
 predefined (list), 80–81
 ranges of, 80
 reference types v., 85
 strings as, 242

ValueCollection class, 422

var keyword, 69, 88, 128, 399, 437, 440

variables, 69–70. See also specific variables

 global, 72
 as keywords, 70
 local, 72
 promoting, to parameters, 48
 renaming, 39–40
 scope of, 71–74
 watching, 52

VB.NET (Visual Basic .NET), 4, 6, 7, 8, 67, 194, 265, 545, 546, 577, 591, 620

VBScript, 545, 546

versions

 .NET CF, 576–577
 .NET Framework, 8–9, 757–759
 of same assembly, 490–491, 492, 494
 Visual Studio, 8, 695

video project. See Media project

View Help File From Last Build button, 776

virtual keyword, 69, 189, 195, 754

virtual methods, 145, 189–190

Vista (Windows), 546, 758

 Administrator and, 493, 495
 FTP service and, 504
 .NET Framework and, 546

Visual Basic .NET (VB.NET), 4, 6, 7, 8, 67, 194, 265, 545, 546, 577, 591, 620

Visual C# Development Settings, 12–13

visual cues, 565

Visual Studio 2008, 11–59

 AJAX-enabled WCF Service template, 741, 743
 components, 17–48
 debugging tools. See debugging
 development settings, 12–14
 resetting, 13–14
 editions, 11–12, 62
 new project creation, 14–16
 Object Browser, 759–764
 O/R Designer, 458, 549
 PhotoViewer, creation of, 506–508. See also PhotoViewer
 shared assemblies in, visibility of, 495
 Signature project and, 676–693
 Silverlight Tools Beta 1 for, 620, 675, 676
 versions, earlier, 8, 695
 WCF Service Library application, 700–710
 WCF ticketing service, 730–741
 Windows Mobile applications and, 578
 Windows Mobile Form in, 579, 580

Visual Studio .NET Help format (HTML Help 2), 773

void keyword, 67, 69, 754

volatile keyword, 69, 299, 754

W

Wacom Intuos Pen Tablet, 675

Wait () method, 309

#warning, 117–118

WAS (Windows Activation Service), 713

watching, 52

WCF (Windows Communication Foundation), 5, 621, 695–745, 758

 ABCs of, 710–711
 addresses, 710, 711
 ASMX Web Services v., 695, 700
 bindings, 710, 711, 712
 communication protocols, 710
 contracts, 710, 711, 712
 messaging patterns, 713
 rationale of, 695
 services. See WCF services

WCF Service Configuration Editor, 714, 724

WCF Service Host, 713, 720, 730

WCF Service Library application

 consuming, 708–710
 creating, 700–708, 730

WCF services

 AJAX and, 741–745
 building

WCF services (continued)

- multiple endpoints, 714–720
- self-hosted, 720–728
- calling, 710
- hosting, 700, 713
- InstanceContextMode property and, 724
- self-hosted, 713, 720–728

WCF Test Client, 703, 704, 705, 706

WCF ticketing service, 728–741

- building, 730–736
- client for, 736–740
- flow chart, 729
- testing, 740–741

WcfServiceLibraryTest, 700, 701

WcfTicketingService project, 730

web applications, ASP.NET. See ASP.NET

Web Development Server, ASP.NET, 696–697, 714

web methods, 700

web pages

- MSDN–Online style, 773
- refreshing, 560

Web Servers, 713

Web Services (ASMX), 5, 534, 545, 758.

See also WCF

- Base64 encoding and, 488
- calling, 293, 710
- compression and, 353
- creation of, 695–700
- request/response communication model, 700, 713, 728
- signature sent to, 687–693
- WCF v., 695, 700

Web Services Description Language (WSDL), 699, 710, 727

WebBrowser control, RSS Reader, 582

WebClient class, 524

WebRequest class, 503, 518, 587

WebResponse class, 503, 587

WebService.asmx, 687

WF (Windows Workflow), 5, 758

where keyword, 69, 272, 275, 755

while keyword, 69, 754

while loop, 99

whitespace

- characters, 262
- regular expression /s, 261
- trimming, from strings, 251

widening, 89

WinBase64 project, 496

WinBase64.exe assembly, 498

window

- Autos, 53
- Designer, 17, 30–31
- Error List, 17, 29–30

- Immediate, 53
- Objects and Timeline, 636, 640, 642, 647
- Output, 30
- Properties, 17, 28–29

Windows Activation Service (WAS), 713

Windows application, development of, 503–543.

See also PhotoViewer

Windows CE Application Manager, 606, 607, 614

Windows Communication Foundation. See WCF

Windows Forms, 5, 28, 30, 125, 126, 758

- BackgroundWorker control in, 315–319
- thread safety in, 303, 312–314

Windows Live Local site, 560

Windows Media Player, 655. See also Media project

Windows Mobile. See also Pocket PCs; RSS Reader application; Smartphones

- application development, 573–616
- emulators, 577, 596–597
- Form, 579, 580
- Mobile 6 Standard (Smartphone), 574, 576
 - .NET CF and, 581
 - RSS Reader application, 580–616
 - SDK, 578, 581
- platform, 574–575
 - Add Web Reference Option and, 698
 - devices for, 574
 - languages for, 577
 - naming conventions, 574
 - .NET CF and, 576–577

Windows Presentation Foundation (WPF), 5, 23, 610, 621, 713

Windows Presentation Foundation/Everywhere (WPF/E), 617. See also Silverlight

Windows Security Alert dialog, 726

Windows Service, 713

Windows Vista. See Vista

Windows Workflow (WF), 5, 758

WindowsApp–Util, 470, 472

WindowsApp–Utils, 477

WindowsMedia.wmv file, 655, 664

WPF (Windows Presentation Foundation), 5, 23, 610, 621, 713

WPF/E (Windows Presentation Foundation/Everywhere), 617. See also Silverlight

Write checkbox, 505

Write () method, 332, 339

WriteAllBytes () method, 327

WriteAllLines () method, 327

WriteAllText () method, 327

WriteByte () method, 332

WriteFile () method, 156

WriteLine () method, 247, 403

write-only properties, 142–143

writing/reading to files, 327–328

Wrox books (query), 452–453
WS endpoint, 714, 716
WSDL (Web Services Description Language), 699, 710, 727
WSDualHttpBinding, 712
WSFederationHttpBinding, 712
ws_GetSignatureCompleted event handler, 691
WSHttpBinding, 706, 712, 714, 716, 719, 720
ws_SaveSignatureCompleted event handler, 690

X

XAML (Extensible Application Markup Language), 620
 Expression Blend 2 and, 635–644
 MediaPlayer project and, 666–667
 Silverlight user interface and, 621, 622, 623–624
XAttribute class, 450
XDocument object, 450, 452, 454
XElement class, 450
XML
 code documentation feature. See documentation comments
 added to Point class definition, 767–771
 template, 767, 768
 documentation tags, 768
 <c>, 768, 771
 <code>, 768, 771
 <example>, 768, 771
 <exception>, 768
 <include>, 768
 <list>, 768
 <para>, 768
 <param>, 768
 <paramref>, 768
 <permission>, 768

<remarks>, 768
 <returns>, 768
 <see>, 768
 <seealso>, 768
 <summary>, 767, 768
 <typeparam>, 768
 <typeparamrefs>, 768
 <value>, 768

LINQ to. See LINQ to XML trees, creating, 450–452

XML serialization, 359, 365–375
 customizing, 372–374
 default constructor for, 374–375
 sample class
 defining, 365–367
 deserializing, 370–371
 serializing, 367–369
 uses of, 375

XmlReader class, 370

XmlSerialize () function, 367, 368, 369

XMLSerializer class, 367, 370, 372

XNamespace element, 456

XPath, 425

XQuery, 425

XSLT, 425, 773

Y

yield keyword, 69, 409, 755

YouTube player, 663. See also MediaPlayer project

Z

ZIndex attribute, 625

