

Index

COPYRIGHTED MATERIAL

Index

Symbols and Numerics

- & (ampersand) for pointers, 341–342**
- ! (exclamation mark) in WF, 1488**
- # (pound sign) in preprocessor directives, 63**
- () parentheses in regular expressions, 220**
- * (asterisk)**
 - in regular expressions, 218
 - for pointers, 341–342
- // (forward slashes) in comments, 27, 60–61**
- /* (slash-asterisk) in comments, 27**
- ; (semicolons) in statements, 27**
- [] (square brackets)**
 - in Array class, 127
 - in regular expressions, 218
- \ (backslash) in regular expressions, 217**
- { } (curly braces)**
 - in C#, xviii
 - in if statements, 42–43
 - initializing arrays and, 122
 - statements and, 27
- ~ (tilde)**
 - in controls, 1314
 - in destructors, 335
- + (plus sign) in DataGridView control, 1069**
- < % @ Register % > directive, 1318–1320, 1324**
- 1-dimensional arrays, defined, 125**
- 3-D features in WPF**
 - basics, 1237–1238
 - triangles. See triangle models (WPF)

A

- abstract classes/functions, 106–107**
- access control lists (ACLs), 599–601, 825–828**
- access modifiers**
 - in .Net languages, 1699
 - for properties, 84–85
- Account class example (generic methods), 236–238**
- account management (Active Directory), 1616–1620**
- Accumulate() method, 237–239**
- ACID properties, transactions, 681**
- acknowledgement queues, 1576–1577**
- acknowledgment messages, 1559**
- AcquireLifetimeToken() method, 1258–1259**
- action panes (VSTO), 1404–1405**
- Activate() method (add-ins), 1255–1256**
- Activator class (add-ins), 1252, 1255–1256**
- Active Directory**
 - administrative tools, 1594–1596
 - characteristics of data, 1592
 - configuration, 1589
 - domain, 1589–1590
 - domain controller, 1590
 - Domain Services Integration, 1558
 - domain tree, 1590
 - Domains and Trusts MMC snap-in, 1594
 - features, 1588
 - forest, 1591
 - global catalog, 1591
 - objects and, 1589
 - programming. See Active Directory, programming
 - replication, 1591–1592
 - schema, 1589, 1592–1593
 - site, 1590
 - Sites and Services MMC snap-in, 1594
- Active Directory, programming, 1596–1611**
 - ADSI, 1596–1597
 - binding. See binding, Active Directory
 - cache, 1605
 - directory entries, updating, 1606
 - new objects, creating, 1605–1606
 - searching, 1608–1611
 - System.DirectoryServices namespace classes, 1598

Active Directory Service Interface (ADSI)

- ADSI Edit, 1594–1595
- basics, 1588
- native ADSI objects, 1607–1608

Active Server Pages (ASP), 19

Active Template Library (ATL), 756–757

active transaction phase, 680

ActiveX controls in Windows Forms, 768–770

activities (WF), 1489–1495. See also custom activities (WF); custom composite activities (WF)

- Activity class, 1489, 1503, 1508
- activity execution model, 1494–1495
- basics, 1487, 1489
- binding parameters to, 1513–1514
- CallExternalMethodActivity, 1492
- composite activities, defined, 1489
- defined, 1487, 1489
- DelayActivity, 1493
- IfExistsActivity, 1490–1491
- ListenActivity, 1493–1494
- ParallelActivity, 1491–1492
- types of, 1502

adapters

- calculator add-in, 1262–1264
- calculator host, 1265–1267

add-ins, 1251–1271

- activation, 1255–1256
- adapter, 1262–1264
- AddInStore class, 1254
- AddInToken class, 1255–1256, 1264, 1270
- AddInToken.Activate(), 1255–1256
- Advanced Calculator, 1271
- application-level (VSTO), 1388, 1390, 1396, 1399–1400
- basics, 1251
- calculator example, 1259–1260
- contract assembly, 1261
- contracts, 1257–1258
- finding, 1252, 1254
- host adapter, 1265–1267
- host application, 1267–1271
- host view, 1265
- implementation, 1264–1265
- issues with, 1251–1253
- pipeline architecture of, 1253–1254

- versioning, 1259
- view, 1261–1262
- when to unload, 1258–1259

add-ons. See add-ins

address operators, 341

admin privileges in applications, 1712–1713

administration queues, defined, 1559

ADO (Active Data Objects), history of, 846

ADO.NET

- assigning transactions to DataContext object, 905
- converting data to XML, 948–954
- converting XML to ADO.NET data, 954–956
- data binding. See data binding and ADO.NET
- database connections. See database connections
- database-specific classes, 847–849
- key generation with SQL Server, 890–892
- namespaces for .NET data access, 846
- overview, 846
- shared classes in System.Data namespace, 847
- tiered development, 889–890
- transactions, 683–684

ADO.NET Entity Framework

- changing information, 1675–1676
- conceptual layer, 1659–1660
- Data Model Designer, 1659
- entities, attaching/detaching, 1677
- entity classes, 1661–1663
- LINQ to Entities, 1678–1679
- logical layer, 1657–1658
- mapping layer, 1660
- object queries, 1670–1674
- object tracking, 1674–1675
- ObjectContext class, 1664–1666
- overview, 1656
- relationships, 1666–1670
- storing entity changes, 1677–1678
- Table per Hierarchy (TPH), 1666–1668
- Table per Type (TPT), 1668–1670
- updates, 1674–1678

ADSI. See Active Directory Service Interface (ADSI)

Advanced Query Syntax (AQS), 1728–1729

aggregate operators, 311, 321–322**aggregates**

- Aggregate template, 997
- defined, 995
- user-defined, 995–997

Ajax. *See also* ASP.NET Ajax

- basics, 1356–1358
- defined, 1355

Ajax Library, 1374–1383

- Application class, 1378–1380
- ASP.NET application services, 1383
- asynchronous Web method calls, 1381–1382
- basics, 1360–1361, 1374–1383
- global utility functions, 1375
- JavaScript, adding to Web pages, 1375
- JavaScript debugging, 1380–1381
- JavaScript OOP extensions, 1375–1378
- PageRequestManager class, 1378–1380

algorithms, encryption/decryption, 592–594**aliases, namespace, 54–55****All Code membership condition, 614****ambient transactions**

- basics, 694–695
- multithreading with, 697–701
- nested scopes with, 695–697

animations, 1228–1236

- basics, 1228–1229
- storyboard, 1235–1236
- timeline, 1229–1232
- triggers and, 1233–1235

anonymous methods (delegates), 190–191**anonymous types, 91–92****ApartmentState property, 765****AppDomain class, 482****Application class**

- Ajax Library, 1378–1380
- methods and properties, 1018–1019

application domains

- CLR code and, 986
- fundamentals, 481–485
- .NET, 12–14

applications

- application configuration (ASP.NET pages), 1308–1309
- application-level add-ins (VSTO), 1388, 1390, 1396, 1399–1400
- basics, 1308
- cache (ClickOnce), 460
- configuration files, 496, 500–503
- defined, 1308
- interacting with (VSTO), 1400–1401
- objects, 1308
- publishing, 459
- requiring admin privileges, 1712–1713
- services (ASP.NET), 1383
- streamlining, 1308
- Web Parts applications components, 1344–1345
- Windows Forms, creating, 1018–1023

architecture

- ASP.NET, 1274
- client-server communications, 1625–1626
- of event logging, 518–519
- of message queuing, 1559–1560
- P2P, 1627–1628
- pipeline (add-ins), 1253–1254
- solutions (VSTO), 1386
- WebRequest/WebResponse classes, 1438–1440

arguments, passing to Main() method, 56–57**arithmetic operators. *See* operators, C#****Array class**

- basics, 127–131
- generic delegates with, 240–243

arrays. *See also* bit arrays (collections)

- Array class, 127–131
- ArrayList class, 224, 255
- ArraySegment <T> struct, 245
- basics, 51–52
- collection interfaces and, 132–133
- collections and, 247
- copying, 128–129
- creating, 127–128
- creating with .Net languages, 1693
- declaring, 121
- defined, 121

arrays. See also bit arrays (continued)

- displaying data from, 1063–1064
- elements, accessing, 123
- enumerations. See enumerations
- initializing, 122
- jagged, 126
- multidimensional, 125
- properties, 127
- QuickArray example, 354–355
- simple, 121–124
- sorting, 129–131
- stack-based, creating, 352–354

as operator, 146

.ascx files, 1313–1314

.asmx extensions (ASP.NET), 1369, 1382

ASP (Active Server Pages). See Active Server Pages (ASP)

ASP.NET

- features of, 19–20
- localization with, 667–669
- overview, 19
- using COM objects from within, 771

ASP.NET Ajax, 1355–1384

- Ajax extensions, 1359–1360
- Ajax Library. See Ajax Library
- basics, 1356–1358
- Control Toolkit, 1361, 1367–1368
- ExtenderControl class, 1372–1374
- overview, 1355
- ScriptManager control, 1368–1370
- support for Web pages, xlvi–xlvii
- UpdatePanel controls, 1370–1372
- UpdateProgress control, 1372
- Web site configuration example, 1365–1368
- Web site simple example, 1362–1364

ASP.NET development, 1311–1353

- basics, 1311–1312
- controls. See custom controls (ASP.NET); user controls (ASP.NET)
- master pages. See master pages (ASP.NET)
- site navigation, 1328–1331
- themes. See themes (ASP.NET)
- Web parts. See Web parts (ASP.NET)

ASP.NET pages, 1273–1310

- ADO.NET and data binding. See data binding and ADO.NET
- application configuration, 1308–1309

- basics, 1273
- overview, 1274
- state management in, 1274–1275
- Web Forms. See server controls (ASP.NET); Web Forms (ASP.NET)

.aspx files, 1274–1275, 1277–1278, 1314

assemblies

- add-ins example, 1260
- application domains and, 481–485
- Assembly Cache Viewer, 487–488, 491
- attributes, 475–478, 1532
- class, 370–371
- code groups, viewing, 616–618
- dynamic loading and, 478–481
- features of, 470
- fundamentals, xlv, 15–16, 469–470
- manifests, 472
- metadata, 470–471
- modules and assemblies, creating, 474–475
- namespaces and, 472
- private, 16, 472–473
- publisher policy, creating, 504–505
- reflection and, 17
- satellite, 473
- shared. See shared assemblies
- structure of, 470–471
- types defined in, 370
- VectorClass assembly example, 364–365
- viewing, 473
- WhatsNewAttributes library assembly, 362–363

asserts, tracing and, 516–517

asymmetric encryption keys, 591–594

AsyncCallback delegate, 537

asynchronous callback, 537–538

asynchronous delegates, 535–538

Asynchronous JavaScript and XML. See Ajax

asynchronous page requests, 1429

asynchronous patterns. See

- BackgroundWorker class; event-based asynchronous pattern**

asynchronous postbacks, 1355

asynchronous programming, 1555

asynchronous reads (messages), 1569

asynchronous Web method calls, 1381–1382

AsyncPostBackTrigger control (ASP.NET Ajax), 1360

ATL (Active Template Library), 756–757
Atom readers, 1643–1645, 1650
atomicity, transaction, 681
attached property (WPF), 1153–1154
Attendees table (Event-Booking example), 1295–1296
attributes
 assembly, 475–478
 AttributeTargets enumeration, 359
 AttributeUsage attribute, 359–360
 COM Interop, 774–777
 custom. *See* custom attributes
 custom controls (Windows Forms), 1049
 Enterprise Services, 1532–1533
 in .NET, 15
 parameters, specifying, 361–362
 properties as XML attributes, 1152
 retrieving attribute data, 927–928
 <siteMapNode> element, 1329
 System.Runtime.InteropServices namespace, 774–775
 transactions (Enterprise Services), 1539–1540
 XAttribute object, 974
authentication and authorization
 Active Directory, 1601
 ASP.NET, 1331
 client application services, 586–591
 declarative role-based security, 585–586
 of forms, 1331–1334
 identities and principals, 583–585
 role-based security, 585
 WebRequest class and, 1428
Authenticode signatures, 486
authority (PNRP IDs), defined, 1630
authorization and authentication. *See* authentication and authorization
auto-implemented properties, 85
automatic transactions (Enterprise Services), 1529, 1539
Autos window (Visual Studio), 432

B

background threads
 basics, 541–542
 thread pools and, 545
BackgroundWorker class
 basics, 571–574
 cancel functionality, 574–575
 progress information, 575–576
backward compatibility, 339
BareTheme directory, 1341–1342
Base class library (BCL), 17, 807, 1361
base keyword, 109
BasicForm Windows project, 413–415
BeginInvoke() method (threading), 535, 537–538, 558, 570
Beginning Regular Expressions (Wrox Press), 215
Beginning XML (Wiley), 921
big-O notation, 294–295
binary code reuse, 225
binary files, reading/writing to, 810–815
BinaryFileReader example, 812–815
BinaryReader/Writer classes, 809–810
binding
 Binding class, 1064–1065, 1078
 BindingContext property, 1077–1078
 early, 752
 late, 752
 method, 752
 WCF, 1477–1480
 WSDL and, 1458
binding, Active Directory, 1598–1604
 authentication, 1601
 directory entries, getting, 1602–1603
 with DirectoryEntry class, 1602
 distinguished name, 1599–1601
 object collections, 1603–1604
 port number, 1599
 protocol, 1598–1599
 server name, 1599
 serverless, 1599
 username, 1601
bit arrays (collections)
 BitArray class, 289–291
 BitVector32, 291–294
bitmap block transfer (BitBit), 1123
BitTorrent, 1625, 1627
BitVector32 structure, 291–294
blittable data types, 755, 990
bool value type, 35

boolean types, 38–39

box, 3-dimensional, 1242–1245

boxing and unboxing

- casts, 170–171
- for converting types, 152–153
- defined, 224

break statements, 49

breakpoints, debugging, 431–432

browsers

- application, creating, 768–770
- Browsable attribute, 1072
- Browser Application (WPF), 1249
- compatibility (ASP.NET Ajax), 1360
- Web browsing, 1430–1431

brushes

- GDI+, 1116–1117
- WPF, 1163–1167

BubbleSorter example (delegates), 184–187

bubbling events, 1177

buffered streams, 810

buffers, defined, 810

building code, defined, 427

business service layer (Enterprise Services), 1528

buttons

- Button class (Windows Forms), 1026–1027
- pin, 426

byte code markers, defined, 816

byte types in C#, 37

C

C ++/CLI language

- mapping syntax from C#, 1681–1709
- mixing native and managed code, 1708–1709

C# language

- basic features, xlviii–xlix
- declaring delegates in, 178–179
- limitations of, xlix
- mapping syntax to VB and C ++/ CLI, 1681–1709
- .NET framework and, xliii–xliv, xlviii, 4
- programming languages and, xlviii–xlix
- significance of, xliii

C# language basics, 25–73

- arrays, 51–52
- comments, adding, 60–62

compiling, 57–58

console I/O, 58–60

enumerations, 50–51

flow control. See flow control

Main() method, 55–57

namespaces, 52–55

as object-oriented program, 25–26

predefined data types. See data types, predefined

preprocessor directives, 63–66

programming guidelines. See programming guidelines (C#)

simple example, 26–29

variables in. See variables

Cab installer Project, 448

cache

- Active Directory, 1605
- application, 460

calculator example (add-ins), 1260–1271

calendars

- control, customizing, 1297–1300
- information, 647–648

callback

- asynchronous, 537–538
- functions, 177

calls

- asynchronous Web method (ASP.NET Ajax), 1381–1383
- CallExternalMethodActivity (WF), 1492–1493, 1499, 1520

camel casing, 69

cancel functionality (BackgroundWorker class), 574–575

Canvas panel container (WPF), 1173–1174

CapsEditor example (GDI+), 1129–1133

captures in regular expressions, 220–222

Cascading Style Sheets (CSS)

- Ajax and, 1356–1357
- ASP.NET themes and, 1339
- master pages and, 1327–1328
- site navigation and, 1331

casing for names, 68–69

caspol.exe (Code Access Security Policy), 614–618

casting

- multiple, 171–175
- between pointer types, 343–344
- pointers to integer types, 343

casts

QueryInterface() method and, 752
 user-defined. See user-defined casts

catalogs, global (Active Directory), 1591**catch blocks**

defined, 379–380
 examples, 382–386
 multiple, 382–385

catching exceptions, 379–381, 385**catching user-defined exceptions, 390–391****CCW (COM callable wrapper), 771****certificates, distributing code with, 631–637****ChangeState() method (threading), 545–547****ChannelFactory class, 1482–1483****character types**

basics, 39
 in C#, 37
 Unicode characters and, 641

characters

grouping with regular expressions, 220
 in regular expressions, 217–218

CheckBox control (Windows Forms), 1028**checked/unchecked operator, 145****CheckedListBox control (Windows Forms), 1029–1031****child domains (Active Directory), 1590, 1609****child elements, properties as (XAML), 1152****ChildrenAsTriggers property, 1370, 1371****classes. See *also* derived classes**

3-D (WPF), 1237–1238
 abstract, 106–107
 account management (Active Directory), 1616–1617
 adapter, 1262–1264
 calculator add-in, 1264
 casts between base and derived, 169–170
 class hierarchy (Windows Forms), 1023
 class library using sockets, 719–722
 class members, pointers to, 346–347
 Class View window, VS 2008, 424
 collections, 250
 commands classes, 1224
 converting casts between, 168–169
 Course Order application, 1570
 for creating syndication feeds, 1644–1645
 data members, 77
 database, 681–683

database-specific (ADO.NET), 847–849
 defining Ajax Library, 1376
 dictionaries, 284–285
 entity, 681–683, 1579–1580, 1661–1663
 event logging, 519–520
 exception, 378–379
 for file system management, 791–792
 function members, 77–78
 generic, creating, 226–230
 generic, features, 231–235
 generic library, 178
 host adapter (add-ins), 1265–1266
 inheritance and, 102
 to integrate Windows Forms and WPF, 1245
 Internet protocols, lower level, 1444
 for IP addresses, 1441–1443
 LINQ to SQL (example), 898–899
 members, 76
 Microsoft.SqlServer.Server namespace, 987–988
 .NET, and file/folder management, 792–794
 .NET base classes, 28
 .NET data access, 846–847
 .NET Framework, 17–18
 .NET Registry, 830–833
 partial, 95–96
 performance monitoring, 528
 principal, 583–585
 sealed, 107
 shape (WPF), 1162
 shared in System.Data namespace, 847
 static, 96–97
 stream-related, 809
 vs. structs, 76, 102
 System.Data.Common namespace, 847
 System.DirectoryServices namespace, 1597, 1598
 System.DirectoryServices.Protocols namespace, 1597, 1621
 System.Net.Sockets namespace, 3
 System.Windows.Forms namespace, 1076
 System.Xml.Serialization, 957
 System.Xml.XPath namespace, 936–942
 timer, 568
 for tracing, 510–511. See *also* tracing
 TriggerAction, 1236
 UpdatePanel control, 1371

classes. See also derived classes (continued)

- user-defined exception, 394–396
- utility (Web programming), 1440–1443
- VSTO, 1385, 1392, 1400
- WPF hierarchy, 1155–1156
- XML, 922–923
- XmlCharacterData, 932
- XmlLinkedNode, 932
- XmlNode, 931–932

clickCountPerSec field, 530

ClickOnce deployment

- application cache, 460
- defined, 444
- File System editor, 460–461
- fundamentals, 458–459
- publishing applications, 459
- security, 460
- settings, 459
- vs. Windows installer, 458

client applications

- building, 1450–1451
- creating with sink object, 781–782
- Enterprise Services, 1432, 1442–1443, 1548–1549, 1553–1554
- integrating WCF with Enterprise Services, 1447–1448
- services, 586–591

clients

- client from Web server installation, 457
- client-server communications, 1625–1626
- client-side code, UDTs from, 994–996
- WCF, 1464–1465, 1482–1483

clipping regions (GDI+), 1100–1102

closed locked/unlocked state (WF), 1509–1511

clouds (P2P)

- defined, 1628
- PNRP, 1630–1631

CLR (Common Language Runtime)

- deploying, 446
- installing/using multiple versions, 506
- overview, 4
- permissions provided by, 603–605

CLS (Common Language Specification). See Common Language Specification (CLS)

Cng (Cryptography Next Generation) API, 593

code

- accessing resources from (WPF), 1180–1181
- ASP.NET model, 1278–1279
- binary reuse (generics), 225
- code-based security, defined, 12
- code-behind feature (ASP.NET), 19
- code bloat (generics), 226
- compiling, 26–27
- displaying from Web pages, 1438
- distributing with certificates, 631–637
- distributing with strong names, 630–631
- downloading source, li
- examples, C#, 26
- generating with XSD, 877–883
- managed, defined, 4
- memory type safe, 14
- mixing native and managed in C ++/ CLI, 1708–1709
- optimization of, 427–428
- sharing, xlv
- syntax basics, 27–29
- UDTs from client-side code, 994–996
- unsafe, and memory management, 339
- unsafe code blocks, 1709

code access security

- basics, 602–603
- code groups. See code groups
- permissions. See permissions
- policy levels, 621–622
- Policy tool, 614–616
- in System.Net.PeerToPeer, 1635

code groups

- changing group permissions, 627–628
- Code Access Security Policy tool, 614–616
- creating, 626–627
- defined, 602
- deleting, 627
- managing, 626
- membership conditions in .NET, 613–614
- viewing assembly, 616–618

<codeBase> element, 497–498

collections, 247–296

- bit arrays. See bit arrays (collections)
- collections initializers (.NET), xlv, 253
- dictionaries. See dictionaries
- generic, 1071–1072
- HashSet, 286–289

interfaces and types, 247–250
 interfaces, arrays and, 132–133
 linked lists, 268–275
 lists. *See* lists
 object (Active Directory), 1603–1604
 performance, 294–296
 queues, 261–266
 sorted lists, 275–278
 stacks, 266–268
 type conversion, 260–261

colors

background, 833–839
 in GDI+, 1114–1116

columns

binding to TextBox, 1075–1076
 constructing, 1072–1074
 custom, 1074–1075
 data, 867–868
 foreign language messages and, 674
 limiting called with query, 910
 naming, 910–911

COM (Component Object Model). *See also***.NET and COM; .NET Framework**

ActiveX controls in Windows Forms, 768–770
 apartments, and threading, 545, 569–570
 vs. C# interfaces, 114
 COM callable wrapper (CCW), 771
 COM client, creating, 778–779, 781–782
 COM client, using .NET component from, 771
 COM+ history, 1528
 COM+ simple application, 1531–1534
 connection points, 765–767, 780–781
 creating component, 756–762
 history of, xliii
 Interop attributes, 774–777
 .NET component, creating, 772
 .NET Framework and, 6
 objects from ASP.NET, using, 771
 primary interop assembly, 764
 registration, 777–778
 runtime callable wrapper (RCW), 762–764
 threading issues, 764–765
 type library, creating, 772–774

ComboBox control (Windows Forms), 1029–1031**command bindings (WPF), 1224–1228****command-line compiler (csc.exe), 26****command link controls (Vista), 1716****Command pattern (data binding), 1090****commands, ADO.NET**

CommandType enumeration, 856
 execute methods, 856–860
 overview, 855–856
 stored procedures, calling, 860–863

comments

within source files, 60
 XComment object, 973
 XML documentation and, 61–62

committable transactions, 687–690**committing transaction phase, 680–681****Common Language Runtime (CLR). *See* CLR (Common Language Runtime)****Common Language Specification (CLS)**

basics, 11
 CLSID, 753

Common Type System (CTS)

basics, 9–11, 36
 predefined value types, 37–39

communications. *See also* message queues;**peer-to-peer (P2P) networking; WCF (Windows Communication Foundation)**

channels (WCF), 1456
 client-server, 1625–1626
 duplex communication (WCF), 1484–1486
 history of technologies, 1455
 WCF and other technologies, 1455–1456

comparison operators (= =)

defined, 154
 overloading, 161–163

compiling

C# files, 57–58
 code, defined, 427
 conditional compilation, 64, 428

components

component load balancing (Enterprise Services), 1529
 Component Services Explorer, 1536–1538
 Component Services MMC snap-in, 691–692
 creating (COM+ application), 1533–1534
 defined, 420
 event-based asynchronous, creating, 576–581
 queued (Enterprise Services), 1530
 WCF and, 1456–1457

compound from filter (LINQ), 313–314

Computer Management MMC snap-in, 1560, 1566

comsvccfg.exe, 1549

conceptual layer (ADO.NET Entity Framework), 1659–1660

Conceptual Schema Definition Language (CSDL), 1659

conditional compilation, 64, 428

conditional operator

fundamentals, 144–145
and .NET languages, 1694

conditional statements

if statement, 42–44
switch statement, 44–45

configuration

Active Directory, 1589
application configuration files, 496, 500–503
/Configuration/Themes/Default.aspx page, 1340–1342

Configuration Wizard, 1296–1297
machine configuration files, 496
VS 2008 configurations, 429–430
Web site example, 1365–1368

Connection property (DataContext objects), 904

connection strings

adding events and, 1300
managing, 850–852

consistency, transaction, 681

consistent bit, setting, 1540

console application

building, 1450–1453
VS 2008 example, 410–412

console I/O, 58–60

constants

basics, 34
classes and, 77
const keyword, 34, 77

constraints

data, 874–876
generics and, 232–234
naming conventions for, 893
setting update/delete, 876

constructors

calling from other constructors, 89–90
defined, 77

defining classes and, 1376
fundamentals, 86–87
in .Net languages, 1689–1690
static, 87–89
for structs, 94–95

constructors of derived classes, 107–112

adding in hierarchies, 109–110
adding to hierarchies, 110–112
defining, 107–109

contacts (P2), 1639–1640

content controls (WPF), 1168–1169

ContentPlaceHolder control (ASP.NET), 1324

<ContentTemplate> element, 1370

ContextMenu attribute, 1089–1090

ContextMenuStrip class (Windows Forms), 1042

continue statements, 49

contract assembly (add-ins), 1261

ContractHandle class, 1259

contracts

add-ins, 1257–1258
WCF, 1467–1471

control class (Windows Forms)

appearance properties, 1024
functionality supported, 1026
size/location properties, 1023–1024
user interaction events, 1024–1025
Windows support, 1025–1026

control statements (C#), 1694–1695

Control Toolkit (Ajax), 1361, 1367–1368

controls

ASP.NET Ajax, 1368–1372
binding to forms, 1077–1078
extender (ASP.NET Ajax), 1372–1374
host (VSTO), 1389, 1392–1394
ribbon menus (VSTO), 1402–1403
Windows Forms. See Windows Forms
WPF. See WPF (Windows Presentation Foundation)

controls (ASP.NET). See also custom controls (ASP.NET); user controls (ASP.NET); Web Parts (ASP.NET)

building custom, 1311
defining default state for, 1314
Login Web server, 1336
navigation, 1328–1329
security, 1337, 1338
WPF, within Windows Forms, 1245–1248

ControlToValidate property, 1288

conventions

- naming, 68–72
- usage basics, 67–68
- used in this book, I–li

conversions

- conversion operators, 311, 322–323
- ConvertAll() method, 240, 242
- type, 149–152

coordinates

- and areas, measuring (GDI+), 1102
- transforms (GDI+), 1137–1138

copying

- arrays, 128–129
- Copy Web tool, 444, 447
- files, 800–805

correlation ID, defined, 1577

Count() extension method, 321

Course Order application example, 1569–1576

- class library, 1570
- message receiver, 1573–1576
- message sender, 1570–1571
- priority/recoverable messages, sending, 1571–1572

CourseManagement sample database, 681–683

covariance/contravariance, 193–194

CPUs

- lock statement and, 551
- race conditions and, 546
- threading and, 533, 542–543

Create() method (MessageQueue class), 1562

CreateInstance() method (arrays), 127–128

Credentials property (WebRequest class), 1428

.cs extensions in C#, 26

CSharpCodeProvider class, 478

CSS (Cascading Style Sheets). See Cascading Style Sheets (CSS)

CTS (Common Type System). See Common Type System (CTS)

CultureAndRegionInfoBuilder class, 677–678

CultureInfo class, 641

cultures, world

- basics, 641

- changing programmatically, 663–665
- creating custom, 677–678
- CurrentCulture/CurrentUICulture, 642–643
- date formatting and, 644–645
- number formatting and, 643–644
- types of, 641–642
- viewing all, 645–649

CurrencyManager object, 1078–1080

CurrentCulture/CurrentUICulture properties, 642–643, 650, 664

Custom Actions editor (VS 2008), 463–465

custom activities (WF), 1495–1508

- activity validation, 1497–1498
- ActivityToolboxItem and icons, 1500–1501
- basics, 1495–1496
- custom composite activities. See custom composite activities (WF)
- themes and designs, 1498–1500

custom attributes

- AttributeUsage attribute, 359–360
- GetCustomAttributes() method, 371
- overview, 358
- parameters, specifying, 361–362
- WhatsNewAttributes example, 362–365, 371–375
- writing, 358–362

custom composite activities (WF)

- attached properties, 1506–1508
- basics, 1502–1503
- composite activities, defined, 1489
- Designer class, adding, 1503–1505

custom controls (ASP.NET)

- .aspx files, 1314
- basics, 1312, 1318–1320
- example, 1320–1323
- vs. user controls, 1318

custom controls (Windows Forms)

- attributes, 1049
- overview, 1048–1049
- TreeView control-based, 1049–1055
- user controls, 1055–1059

custom hosting (WCF), 1480–1481

custom interfaces, 751

custom objects (Northwind database)

- creating (Customer table), 908
- and LINQ, 908–910
- O/R Designer and, 912–914

custom resource managers, 703–709

customizing Office applications. *See* **Visual Studio Tools for Office (VSTO)**

D

data

- accessing (.NET), xliv
- classes, defined, 420
- columns, 867–868
- constraints, 874–876
- contracts (WCF), 1467–1468, 1579–1580
- Data Model Designer (ADO.NET Entity Framework), 1659
- display controls (ASP.NET), 1287–1288
- displaying. *See* DataGridView control
- members (classes), 77
- passing to threads, 540–541
- querying (XQuery), 1008–1010
- readers, 863–865
- relational, and XML data type, 1003
- relationships, 873–874
- rows, 868–871
- service layer, 1528
- sets, populating, 883–884
- templates and WPF binding, 1214–1217
- triggers and data integrity, 1002
- XML, and tables, 1003–1005

data access (Visual Studio .NET), 1080–1092

- data, selecting, 1084
- data source, updating, 1084
- database connections, creating, 1080–1084
- pop-up menu for rows example. *See* pop-up menu for rows

data access with ADO.NET. *See also* ADO.NET

- commands. *See* commands, ADO.NET
- data adapters, updating with, 885–887
- data reader, 863–865
- data sets, populating, 883–884
- database connections. *See* database connections
- DataSet class. *See* DataSet class
- overview, 845
- persisting DataSet changes, 884–888
- XML output, writing, 887–889
- XML schema definitions (XSD), 877–883

data adapters

- populating DataSet class with, 883–884
- updating with, 885–887

data binding

- ADO.NET, 1295–1307
- DataBindings property, 1076
- DataGridView control, 1061
- defined, 1075
- objects, 1076–1080
- simple, 1075–1076

data binding and ADO.NET

- basics, 1295, 1302–1303
- event-booking application. *See* event-booking application
- templates, data display with, 1303–1304
- templates, using, 1305–1307

data binding and WPF

- binding modes, 1200
- data templates, 1214–1217
- list binding, 1208–1211
- list items, adding dynamically, 1213–1214
- object data provider, 1206–1207
- overview, 1199–1200
- simple object binding, 1203–1206
- strings for list binding, 1212–1213
- validation. *See* validation, binding with XAML, 1200–1203
- to XML, 1217–1219

data sources

- Data source controls (ASP.NET), 1287
- overview, 1063
- updating, 1084

data tables

- basics, 866–867
- data columns, 867–868
- data rows, 868–871
- deleting rows, 887
- inserting rows, 885–886
- schema generation, 871–873
- updating existing rows, 887

data types. *See also* XML data type

- blittable, 755, 990
- COM and .NET related, 756
- immutable, 205
- nonblittable, 756
- reference for memory management, 331–333
- registry entries, 830

- value for memory management, 330–331
- VARIANT, 753
- data types, predefined, 34–42**
 - boolean types, 38–39
 - character types, 39
 - CTS types, 36
 - decimal types, 38
 - floating-point types, 38
 - integer types, 37
 - object types, 40
 - reference types, 35–36, 40–42
 - string types, 40–42
 - value types, 35–39
- database connections**
 - connection strings, 850–852
 - efficient use of, 852–854
 - overview, 849–850
 - transactions and, 854–855
 - try . . . catch . . . finally blocks, 852
 - using block statement, 852–854
- databases**
 - classes, 681–683
 - column naming conventions for, 893
 - connections, creating (Visual Studio .NET), 1080–1084
 - database-specific classes, 847–849
 - event-booking application, 1295–1307
 - querying with LINQ, 914–920
 - table naming conventions, 892
- DataContext object**
 - assigning ADO.NET transactions to, 905
 - creating, 911–912
 - LINQ to SQL and, 903–906
- DataContext property, 1205**
- DataGrid control, 1061**
- DataGridView control, 1061–1072**
 - arrays, displaying data from, 1063–1064
 - basics, 1061
 - class hierarchy, 1072–1075
 - DataSource property overview, 1063
 - DataTable, displaying, 1065
 - displaying DataSet class data, 1068–1070
 - displaying DataView data, 1065–1068
 - displaying DataView Manager data, 1070
 - displaying generic collections, 1071–1072
 - displaying tabular data, 1062–1063
 - IListSource/IList interfaces and, 1071
- [DataMember] attribute, 1468**
- DataRow version values, 869**
- DataRowState enumeration, 870**
- DataSet class**
 - data constraints, 874–876
 - data relationships, 873–874
 - data tables. *See* data tables
 - displaying data from, 1068–1070
 - overview, 865–866
 - populating from XML, 884
 - populating with data adapter, 883–884
- DataSource property. *See also* DataGridView control**
 - DataTable and, 1065
 - overview, 1063
- DataTable, displaying in DataGridView control, 1065**
- DataView, displaying data from, 1065–1068**
- DataView Manager, displaying data in, 1070**
- DataViewRowState, 1066–1067**
- DateTimePicker control (Windows Forms), 1031**
- DCOM Proxy, MSMQ, 1558**
- DDX (Dialog Data Exchange), 1075**
- dead-letter queues (messages), 1559**
- deadlocks (threading)**
 - basics, 548–549
 - ReaderWriterLockSlim, 564
- debugging**
 - in C#, overview, 430–434
 - debug and release builds, 427
 - Debug configuration (Visual Studio), 429
 - Debug Location toolbar, 540
 - debugger symbols, 428
 - design-time, 418
 - drawing routines, 1106–1107
 - JavaScript, 1380–1381
 - ServicedComponent class, 1549
 - threads, 540
 - tracing and, 509–510
 - XSLT, 947–948
- decimal types, 38**
- declarative behavior (WCF), 1456**
- declarative permissions, 606**
- declarative role-based security, 585–586**
- declaring**
 - arrays, 121–122
 - methods, 78
 - variables, 29

deep comparisons, 162

default keyword (generic classes), 231–232

default values (generic classes), 231–232

#define/#undef preprocessor directives, 63

DelayActivity (WF), 1493

delaying assembly signing, 492–493

delegates

- anonymous methods, 190–191
- asynchronous, and threading, 535–538
- BubbleSorter example, 184–187
- covariance/contravariance, 193–194
- declaring in C#, 178–179
- defined, 238
- fundamentals, 177–178
- GetAString example, 179–182
- Lambda expressions, 191–193
- MathOperations class example, 182–184
- multicast, 187–190
- in .Net languages, 1703–1704

delegates, generic

- defined, 238
- methods called by, 238–240
- setting delete constraints, 238–243
- using with Array class, 240–243

deleting

- ACLs from files, 827–828
- code groups, 627
- delete constraints, setting, 876
- Delete() method (files), 794, 800–801
- delete/update constraints, setting, 876
- event logs, 525
- files, 800–805
- records, 861
- rows in data tables, 887

Demo class, lock statement and, 552

dependencies

- dependency property (WPF), 1152–1153
- dependent transactions, 692–693
- version, 470

deployment

- ClickOnce. See ClickOnce deployment
- COM+ assemblies, 1534–1535
- deploying .NET runtime, 446
- deployment projects, 444
- designing for, 443
- installer project types. See installer project types
- no-touch deployment (NTD), 457–458

- options overview, 444
- requirements, 444–445
- simple examples, 446

dereference operators, 341

derived classes

- calling base versions of, 107
- constructors of. See constructors of derived classes

design-time debugging, 418

design view editor (VS 2008), 402

Design View window (VS 2008), 419–422

designer/developer cooperation (WPF), 1154–1155

Designer, WF

- adding, 1503–1505
- basics, 1487, 1526
- themes and designs, 1498–1500

destructors

- for freeing unmanaged resources, 335–336
- implementing, 337–339

developer/designer cooperation (WPF), 1154–1155

device context (DC) objects (GDI+), 1095–1096

device coordinates (GDI+), 1113–1114

DHTML (Dynamic HTML), 1357–1358

diagnostics, WCF example, 1466–1467

dialog boxes, installation, 462–463

Dialog Data Exchange (DDX), 1075

dictionaries

- basics, 278
- big-O notation and, 295
- classes, 284–285
- example of, 280–283
- key types, 278–280
- lookup, 283–284

Diffie Hellman algorithm, 594–598

directories. See also Active Directory

- add-in discovery, 1254
- defined, 792
- Directory/DirectoryInfo classes, 792–793
- reading ACLs from, 825–826
- securing, 1336–1339
- structure in Vista, 1715–1716
- themes and, 1340

directory services

- account management, 1616–1620

architecture of Active Directory. See Active Directory

- Directory
- basics, 1587
- Markup Language (DSML), 1620–1623
- programming Active Directory. See Active Directory, programming
- user objects, searching for, 1611–1616

dirty reads, transaction, 701

discovery

- add-ins, 1252, 1254
- of clients (P2P network apps), 1628, 1639

discretionary access-control list (DACL), 599

dispatch interfaces, 751–752

DisplayFile() method, 814–815

DisplayTabularData example, 1062–1063

DisplayText example (GDI+), 1124–1125

DisplayTree() method, 325–327

Dispose() method

- IDisposable interface, 336–338, 1701
- SelfPlacing Window example and, 836

distinguished name (Active Directory), 1599–1601

distributed solutions. See Enterprise Services

distributed transactions

- Distributed Transaction Coordinator (DTC), 686, 691, 1530, 1548
- Enterprise Services, 1530

DllImport properties, 784–785

DNS

- class, 1442–1443
- names, 1441
- servers, 1441

do . . . while loops, 48

DockPanel container (WPF), 1174–1175

Document Object Model (DOM). See DOM (Document Object Model)

documents

- calculating sizes of (GDI+), 1134–1135
- document-level customizations (VSTO), 1387–1390, 1394–1395
- Document Manager Console project, 231–234
- DocumentManager<T> class, 231–234
- interacting with (VSTO), 1400–1401
- XDocument object, 969
- XML. See XML documents

DOM (Document Object Model)

- Ajax and, 1356–1357
- implementation in .NET, 931–932

domains

- Active Directory and, 1589–1590
- controllers, 1588, 1590
- defined, 1589
- trees, 1590

done bit, setting, 1540**DoTheTask, 550–551, 554****DoubleAnimation class, 1228, 1229****downlevel logon (Active Directory), 1601****downloading, 1424. See also Web sites, for downloading**

- files, with WebClient class, 1424
- Web pages (example), 1426

drawing (GDI+)

- DrawingBrush (WPF), 1165
- scrollable windows, 1107–1113
- shapes and lines, 1096–1099, 1118–1120
- text, 1123–1125

drive information, reading, 822–824**DriveInfo class, 822****DSA (Digital Signature Algorithm), 594****DSML (Directory Services Markup Language), 1620–1623****dual interfaces, 752****duplex communication (WCF), 1484–1486****durability, transaction, 681****Dynamic HTML (DHTML), 1357–1358****dynamic loading (assemblies), 478–481****dynamic resources (WPF), 1181–1182****dynamic XML documents, querying, 976–978****DynamicResource markup extension, 1181–1182****E****early binding, 752****ECDiffieHellman algorithm, 594–598****ECDSA algorithm, 594****elements**

- operators, 311
- XElement object, 969–970

elements, lists

- accessing, 254–256
- adding, 253–254
- inserting, 254
- removing, 256–257

#elif preprocessor directives, 64

Ellipse class (animations), 1229

#else preprocessor directives, 64

EmployeeId struct, 280

encryption

fundamentals, 591–594

key exchange/secure transfer, 596–599

signatures, 594–596

#endif preprocessor directives, 64

EndOfStreamException class, 379

endpoints. See also MEX (Metadata Exchange)

endpoints

for VBA code (VSTO), 1389

WCF binding, 1479

#endregion preprocessor directives, 65

Enterprise architecture and C#, 22–23

enterprise policy levels (security), 621–623

Enterprise Services, 1527–1554

automatic transactions, 1529

client application, 1538

Component Services Explorer, 1536–1538

contexts, 1529

deployment, 1534–1535

distributed transactions, 1530

example application. See Enterprise Services application

history of, 1528

integrating WCF with. See WCF, integrating with Enterprise Services

loosely coupled events, 1530–1531

.NET, 684–685

object pooling, 1530

overview, 1527–1531

queued components, 1530

role-based security, 1530

simple COM+ application, 1531–1534

transactions, 1539–1540

where to use, 1528–1529

Enterprise Services application, 1540–1549

client application, 1548–1549

entity classes, 1541–1543

OrderControl component, 1543

OrderData component, 1544–1546

OrderLineData component, 1546–1547

entities (ADO.NET Entity Framework)

attaching/detaching, 1677

classes, 681–683, 1661–1663

LINQ to Entities, 1678–1679

storing changes, 1677–1678

Enumerable class, 305, 323

enumerations

basics, 50–51, 133

defining with .NET languages, 1686

enumerating font families example, 1126–1128

enumerating messages, 1568–1569

foreach statements and, 134

IEnumerator interface, 134

yield statements and, 134–139

Equals() method

key types, 279–281

purpose of, 97

static Equals() method, 154

errors and exceptions

catching exceptions, 379–381, 385

client-side error messages, 1369–1370

in data binding, 1219–1222

#error preprocessor directives, 65

ErrorMessage property, 1288

ErrorProvider component (Windows Forms), 1031–1032

exception classes, 378–379

handling exceptions, 389

handling in VB 6, 377

handling in WCF, 1476–1477

handling in .NET and COM, 754

handling with exceptions, overview, 14–15

modifying exception types, 388–389

multiple catch blocks, 382–385

nested try blocks, 387–389

overview, 377

SimpleExceptions example, 382–385

System.Exception properties, 386

unhandled exceptions, 386–387

user-defined exceptions. See user-defined exceptions

escape codes, regular expressions, 215

escape sequences

of chars, 39

regular expressions, 217–218

event-based asynchronous pattern

BackgroundWorker class. See

BackgroundWorker class

basics, 570–571

component, creating, 576–581

event-booking application

- adding events to database, 1300–1301
- binding to database, 1296–1297
- calendar control, customizing, 1297–1300
- database, 1295–1296
- event handling**
 - delegates and, 195–197, 244
 - FileProperties example, 797–800
 - .NET and COM, 754–755
 - simple controls, 1316–1318
 - VSTO, 1397–1398, 1400
 - WPF, 1176–1177
- event logging**
 - architecture of, 518–519
 - basics, 509
 - classes, 519–520
 - event log listeners, 526–527
 - event logs, writing, 522
 - event sources, creating, 521–522
 - Event Viewer, 509, 517–519, 522
 - log entry example, 517–518
 - resource files, 522–526
 - Windows Services, 746
- events**
 - adding to event-booking application, 1300–1301
 - Ajax Library, 1378–1379
 - BackgroundWorker class, 571, 572
 - bubbling, 1177
 - defined, 77
 - defining, 197–201
 - defining in .NET languages, 1705–1706
 - event keyword, 562
 - event triggers (animations), 1229, 1233–1236
 - EventArgs-based object, 198–201
 - EventHandler delegate, 196
 - EventHandler <EventArgs>, 244–245
 - EventLog class, 520, 522
 - EventSourceCreationData, 521–522
 - fundamentals, 194–195
 - loosely coupled (LCE), 1530–1531
 - receiver view of, 195–197
 - synchronization and, 562–564
 - table (MeetingRoomBooker example), 1296
 - tunneling, 1177
 - user interaction (Control class), 1024–1025
 - Web server controls (ASP.NET) and, 1284
- Excel**
 - host items/controls for (VSTO), 1393–1394
 - XML data type and, 1003
- exceptions**
 - catching, 379–381, 385
 - classes, 378–379
 - debugging and, 433–434
 - error handling with, 14–15
 - handling for binding validation, 1219–1221
 - handling in different places, 389
 - handling in .Net languages, 1697–1998
 - modifying types, 388–389
 - throwing, 380
 - unhandled, 386–387
 - user-defined. *See* user-defined exceptions
- Execute method (WF), 1489, 1492, 1494–1496**
- execute methods**
 - ADO.NET commands, 856–860
 - ExecuteNonQuery() method, 857
 - ExecuteQuery<T>() method, 903–904
 - ExecuteReader() method, 857–858
 - ExecuteScalar() method, 858–859
 - ExecuteXmlReader() method, 859–860
- explicit type conversions, 150–152**
- express messages, defined, 1559**
- Expression Blend, 1154**
- expressions**
 - Expression classes, 325
 - expression placeholder, 1303
 - expression trees (LINQ), 324–327
 - Lambda. *See* Lambda expressions
 - query, 914–916
 - regular. *See* regular expressions
- extender providers**
 - .NET, 1508
 - WF, 1506, 1508
- ExtenderControl class (ASP.NET Ajax), 1372–1374**
- extensibility, WCF and, 1456**
- Extensible Application Markup Language (XAML). *See* XAML (XML for Applications Markup Language)**
- Extensible Markup Language (XML). *See* XML (Extensible Markup Language)**

extension methods

creating with C# and VB, 1692
fundamentals, 99–100, 304–306

extensions

Ajax extensions, 1359–1360
document-level customizations (VSTO), 1394
class, 327
JavaScript OOP, 1375–1378
markup (XAML) and, 1362

external safety level, 986

external service, defined (WF), 1511

F

families, font (GDI+), 1125–1126

fat-client applications, 21

feeds, syndication, 1647–1651

Fiddler tool, 1381

fields

basics, 73
defined, 77
read-only, 90–91
scope clashes for, 33–34

file dialogs (Vista), 1721–1722

files

File/FileInfo classes, 792–794
File Search (Launch Conditions editor), 465
file-sharing communications technologies, 1627
File System editor (VS 2008), 460–461
File Transfer Protocol (FTP), 1428–1429
File Types editor (VS 2008), 461–462
FileMode/FileAccess/FileShare enumerations, 810–811
FileStream class, 809, 810–815
FileStream instances, 810
opening in WPF, 1226
resource files (event logging), 522–526
resource files (localization), 651, 653–658
security of, 825–828
/target switches for specifying, 57

files, managing, 791–805

basics, 791–792
file browser example, 796–800

FilePropertiesAndMovement example, 801–805
manipulating files, 800–805
.NET classes and, 792–794
Path class and, 794–795

files, reading and writing to, 805–821

binary files, 810–815
reading, 805–807
streams and, 808–810
text files, 815–821
writing to, 807–808

filtering

Active Directory searching and, 1608
compound from filter, 313–314
Filter property, 515
LINQ queries, 311–314
operators, 310
with query expressions, 916
rows, 1065–1067
tracing and, 510, 515–516

Finalize() method

destructors and, 335–336
overriding, 1702–1703
purpose of, 97

finalizers, defined, 77

finally blocks

defined, 379–380
examples, 382–386

FindAll() method, 240, 242

first in, first out (FIFO) (queues), 261

fixed keyword (pointers), 346–347, 351

flags, localization and, 659–660, 663

floating-point types, 38

floats values, 166

flooding, defined (P2P), 1628

flow control

conditional statements, 42–45
jump statements, 49–50
loops, 45–49

FlowLayoutPanel container (Windows Forms), 1037–1038

folders

defined, 792
.NET classes and managing, 792–794
SpecialFolder enumeration (Vista), 1715

folding editor (VS 2008), 416–418

fonts and families (GDI+), 1125–1128

for loops

- basics, 45–47
- jagged arrays and, 126

for statement in .Net languages, 1696**foreach loops**

- basics, 48–49
- interfaces and, 115

ForEach() method, 240–241**foreach statements**

- accessing array elements and, 123
- enumerations and, 133
- IEnumerable interface and, 134
- iterating through collections with, 135
- in .Net languages, 1697

foreground threads, 541–542**foreign keys, setting, 875–876****forests (Active Directory), 1591****format names (message queues), 1564–1565****format strings**

- fundamentals, 209–214
- for predefined types, 59

FormattableVector example, 212–214**formatting messages**

- message formatter, 1566–1567
- names for opening message queues, 1564–1565

forms

- ActiveX controls in Windows Forms, 768–770
- controls, binding to, 1077–1078
- Forms Authentication (ASP.NET), 1331–1334
- Windows. See Windows Forms

Formula-1 racer example. See LINQ (Language Integrated Query)**FormView control (Web server controls), 1305–1306****FrameworkTemplate base class, 1184****FTP (File Transfer Protocol), 1428–1429****functions**

- abstract, 106–107
- base versions, calling, 106
- function calls, avoiding, 70–72
- function members, 77–78
- user-defined (SQL), 1000–1001
- virtual base class, 104

Futures (ASP.NET Ajax), 1358**G****GAC (global assembly cache). See global assembly cache (GAC)****garbage collection**

- basics, 11–12
- memory management and, 333–334

Garrett, Jesse James, 1356**GDI+ (graphical device interface)**

- brushes, 1116–1117
- CapsEditor example, 1129–1133
- clipping regions, 1100–1102
- colors, 1114–1116
- coordinate transforms, 1137–1138
- coordinates and areas, measuring, 1102
- debugging, 1106–1107
- device context (DC) objects, 1095–1096
- device coordinates, 1113–1114
- drawing scrollable windows, 1107–1113
- drawing shapes and lines, 1096–1099, 1118–1120
- drawing text, 1123–1125
- fonts and families, 1125–1128
- GDI+ namespaces, 1095
- graphics display modes, 1115–1116
- GraphicsUnit enumeration, 1114
- images, displaying, 1120–1122
- images, manipulating, 1123
- Invalidate() method, 1133–1134
- item and document sizes, calculating, 1134–1135
- OnPaint() method, 1099–1100, 1135–1137
- overview, 1094
- page coordinates, 1113–1114
- painting shapes, 1099–1100
- pens, 1116–1118
- Point/PointF structs, 1102–1103
- printing/print previewing, 1141–1146
- Rectangle/RectangleF structs, 1105
- red-green-blue values, 1114–1115
- Region class, 1106
- safety palette, 1116
- Size/SizeF structs, 1103–1104
- user input, responding to, 1138–1141
- world coordinates, 1113–1114

generation operators, 311, 323–324**generic library classes, 178**

generics

- ArraySegment <T> struct, 244–245
- classes, creating, 226–230
- collections, displaying, 1071–1072
- constraints and, 232–234
- default values and, 231–232
- delegates. *See* delegates, generic
- Document Manager Console project, 231–234
- EventHandler <EventArgs>, 244–245
- inheritance and, 234–235
- interfaces, 235–236
- keywords and, 231
- methods, 236–238
- Nullable<T> type, 243–244
- overview, 223–226
- static members and, 235
- supported in .NET languages, 1707–1708

GetAString example (delegates), 179–182

GetCurrent() method, 1368–1369

GetCustomAttributes() method, 370–371

GetEnumerator() method

- enumerations and, 133–135
- generic classes and, 227

GetHashCode() method

- key types, 278–280
- purpose of, 97

GetMethods() (System.Type), 367

GetType() method, 98, 365

global assembly cache (GAC)

- adding publisher policy assemblies to, 505
- fundamentals, 487–489

global catalogs (GC) (Active Directory), 1591

global clouds, 1631

global utility functions (Ajax Library), 1375

globalization

- basics, 639–640
- cultures and regions, 641–645
- cultures in action, 645–649
- defined, 639
- sorting, 650–651
- Unicode issues, 640–641

globally unique identifiers (GUID) (Active Directory), 1601

Globals class (VSTO), 1397

Go To Definition option (VS 2008), 424

goto statements, 49

graphics

- class, 1096. *See also* GDI+ (graphical device interface)
- display modes, 1115–1116
- GraphicsUnit enumeration (GDI+), 1114

graphs, defined (P2P), 1628

Grid container (WPF), 1175–1176

grouping

- database items with queries, 918–919
- GroupBy() method, 315
- operators, 310
- query results (LINQ), 315–317

groups

- account management, 1619
- in regular expressions, 220–222

GUIDs (Active Directory), 1601

H

hand-coded schema, 871–873

handler function (Windows Services), 717–718

handler methods (Windows Services), 728–729

handling exceptions. *See* exceptions

hash algorithms, 593

hash tables. *See* dictionaries

hashing, 1000–1001

HashSet<T> collection class, 286–289, 295

hatch brushes (GDI+), 1117

headers

- headered content controls (WPF), 1170–1171
- headered items controls (WPF), 1171–1172
- HTTP, 1427–1428

Hello World application (WF), 1488–1489

HelpProvider component (Windows Forms), 1032

hiding methods, 104–105

hierarchy

- data binding objects, 1076–1077
- of data in Active Directory, 1588
- DataGridView class, 1072–1075
- pop-up menu classes, 1085
- XML data, 1081

hives. *See* registry

host classes/controls (VSTO), 1389, 1392–1394

hosting applications (add-ins)

- host adapter, 1265–1267
- host application, 1267–1271
- host view, 1265

hosting workflows (WF), 1524–1525**hosts**

- basics in WCF, 1480–1482
- for components and services (WCF), 1456, 1463–1464

HTML (HyperText Markup Language)

- Ajax and, 1356, 1357
- server controls (ASP.NET pages), 1279–1280

HTML pages, displaying output as

- basics, 1429–1430
- Internet Explorer instances, 1432
- Internet Explorer-type features, 1432–1437
- requested-page code, displaying, 1438
- simple Web browsing, 1430–1431
- WebBrowser control for printing, 1437
- WebRequest/WebResponse hierarchy, 1438–1440

HTTP (HyperText Transfer Protocol)

- headers, 1427–1428
- <httpHandlers> (ASP.NET Ajax), 1366
- <httpModules> (ASP.NET Ajax), 1366
- MSMQ, 1558

Hungarian notation, defined, 68**HyperText Transfer Protocol (HTTP). See HTTP (HyperText Transfer Protocol)****I****ICompletedEvents interface, 755****IAsyncResult interface**

- asynchronous callbacks and, 537–538
- BackgroundWorker class and, 571
- polling and, 535–536
- wait handles and, 536, 558

ICollection interface

- arrays and, 132
- collections and, 248, 250

IComparable interface, 129–130**ICompletedEvents interface, 755****IConnectionPointContainer interface, 755****icons**

- custom activities (WF), 1500–1501
- ribbon menus (VSTO), 1403–1404

IContract interface, 1257**IDataErrorInfo interface, 1221–1222****identifiers**

- of controls, 1280
- rules governing, 66–67

identities, user, 583–585**IDictionary interface, 249****IDispatch interface, 751****IDisposable interface**

- basics, 114–115
- for freeing unmanaged resources, 336–339
- implementation, 1701

idle workflows (WF), 1517**IDs, PNRP, 1630****IEnlistmentNotification interface, 704****IEnumerable interface**

- arrays and, 132, 133
- basics, 133
- collections, 248
- generic classes, and, 229

IEnumerator interface, 133–134**IEquality.Equals() method, 279****#if preprocessor directives, 64****if statement**

- basics, 42–44
- in .NET languages, 1694

IfElseActivity (WF), 1490–1491**ICollection**

- arrays and, 132–133
- interface, 249, 1701

ICollectionSource interface, 1071**ILookup<TKey> interface, 249****ImageBrush (WPF), 1165****ImageList component (Windows Forms), 1033****images (GDI+)**

- displaying, 1120–1122
- manipulating, 1123

immutable data types, 205**implementation inheritance**

- abstract classes and functions, 106–107
- base versions of functions, 106
- basics, 101–102, 103
- constructors of derived classes. *See* constructors of derived classes
- hiding methods, 104–106
- vs. interface inheritance, 101–102
- sealed classes and methods, 107
- virtual methods, 104

implicit permissions, 610

implicit type conversions, 149–150

importing

- imports keyword, 1682
- Windows Forms ActiveX Control Importer, 768

Increment() method, 530

indexers

- accessing array elements and, 123
- accessing collection classes and, 255
- accessing list elements and, 254–255
- BitArray class, 290
- defined, 78
- IDataErrorInfo interface and, 1222
- searching lists and, 257–258

indexes

- filtering with, 312
- XML, 1010–1011
- XML queries and, 1009

indexing arrays, 125

indirection operators, 341

InfoPath form templates (VSTO), 1388

Information Server (IIS), 1274

inheritance, 101–120

- Ajax Library, 1376–1377
- generics and, 234–235
- implementation. See implementation inheritance
- interfaces. See interfaces
- modifiers, 112–114
- in .Net languages, 1699–1701
- structs and, 94
- types of, 101–102

initializing

- arrays, 122
- collection initializers, 253
- InitializeComponent() method, 422
- initializers, object (C# and VB), 1691
- variables, 29–30

inlining, 85

input strings (regular expressions), 216

insertion, record, 862–863

installer project types

- client from Web server, 457
- creating installers, 449
- no-touch deployment (NTD), 457–458
- overview, 448
- SimpleAppSolutionSetup example, 454–455
- SimpleClientApp example, 449–454

- SimpleWebApp example, 455–456

- Windows installer, 449

installing

- installer classes (Windows Services), 730–734
- installer packages (Component Services Explorer), 1535
- installutil.exe utility (Windows Services), 734
- message queues, 1585
- shared assemblies, 491

instances

- defined, 76
- Internet Explorer, launching, 1432
- WF, 1487

int variable, 36–37

Int32 number type, 280

integer types

- basics, 37
- casting pointers to, 343
- user-defined. See enumerations

integers, pointer arithmetic and, 344–345

interactive services (Windows Services), 745–746

interfaces

- basics, 114–115
- collections, 132–133, 247–250
- COM and, 114
- custom, 751
- defining and implementing, 115–118
- defining with .NET languages, 1685–1686
- derived, 118–120
- dispatch, 751–752
- dual, 752
- generic, 235–236
- interface inheritance, 101–102
- support in .NET, 7–8
- used with contracts, 1257–1258

Interlocked class (synchronization), 555–556

Intermediate Language (IL)

- application domains, 12–14
- assemblies, 15–17
- attributes, 15
- Common Language Specification (CLS), 11
- Common Type System (CTS), 9–11
- error handling with exceptions, 14–15
- garbage collection, 11–12
- object orientation/interface support, 7–8
- security, 12

strong data typing, 8–11
 value/reference types, distinguishing, 8

internal modifier, 112, 114

Internet, accessing, 1423–1453
 lower-level protocols. See lower level protocols
 (network communications)
 output as HTML pages. See HTML pages,
 displaying output as
 utility classes, 1440–1443
 WebClient class, 1424–1426
 WebRequest/WebResponse classes,
 1426–1429

Internet browser application, creating, 768–770

Internet Explorer
 as hosting application, 1251
 IE-type features, creating, 1432–1437
 launching instances of, 1432
 running Windows Forms controls in, 782–783
 toolbars, creating, 1433–1437
 Web browser control and, 1430

Interop attributes, COM, 774–777

interoperability. See also .NET and COM
 language, 5, 9–11
 VBA, 1415–1418

Invalidate() method (GDI+), 1133–1134

invariant cultures, 641–642, 648

invoking methods, 79–80

IP addresses
 basics, 1441
 classes for, 1441–1442

IPHostEntry class, 1442

IResourceReader interface, 659, 674–676

is operator, 146

IsBackground property (threads), 541–542

isolated storage
 reading/writing to, 839–844
 SelfPlacingWindow example and, 834

isolation of add-ins, 1252

isolation (transactions)
 defined, 681
 IsolationLevel enumeration, 702–703
 levels, 701–703, 855

ISupportErrorInfo interface, 754

items controls (WPF), 1171

<ItemTemplate> for data display, 1303

iterator blocks, 135

IUnknown interface, 751

J**jagged arrays, 126****JavaScript**

adding to Web pages, 1375
 Ajax programming and, 1356, 1357
 client-side references, 1369
 debugging, 1380–1381
 JavaScript Object Notation (JSON),
 1458, 1460
 OOP extensions, 1375–1378
 references, 1369

joins

clauses, LINQ and, 317–318
 operators, 310
 performing with queries, 916–918

journal queues, defined (messages), 1559**JSON (JavaScript Object Notation). See
 JavaScript Object Notation (JSON)****jump statements, 49–50****K****Kerberos standard, 1588****key generation with SQL Server, 890–892****keys, registry, 830–831****keywords**

to avoid, 70–72
 C# reserved keywords, 66–67
 generics and, 231
 inheritance, 1699–1700
 names of, 70–72
 .NET classes and, 28

L**labels**

control (Windows Forms), 1033
 defined, 49

Lambda expressions

accessing list elements and, 256
 delegates types and, 191–193
 LINQ and, 306–307
 threading and, 538, 539

**Language Integrated Query Framework
 (LINQ). See LINQ (Language
 Integrated Query)**

language interoperability

- IL and, 5
- strong data typing and, 9–11

languages

- custom cultures and, 677
- Intermediate Language, xliv
- interoperability of, xliv
- multiple, 641
- outsourcing translations, 666–667
- programming with ASP.NET and, 668
- resource files and, 653
- resource messages and, 665–666
- table for storing messages and, 674
- Unicode issues and, 640
- Visual Studio 2008 and, 663

last in, first out (LIFO) containers, 266

late binding, 752

Launch Conditions editor (VS 2008), 465–466

layers of functionality (Ajax Library), 1360–1361

layout containers (WPF), 1172–1176

LCE (loosely couple events), 1530–1531

Length property, array class, 127

licenses, trust, 460

lights in 3-D sample, 1240

Lightweight Directory Access Protocol (LDAP), 1588

#line preprocessor directives, 65

LinearGradientBrush (WPF), 1164, 1240–1242

lines, drawing (GDI+), 1118

link local clouds, 1631

LinkedList/LinkedListNode classes

- generic classes and, 226–230
- linked lists and, 268–269
- LinkedList<T> class, 268–269, 295

LINQ (Language Integrated Query)

- aggregate operators, 321–322
- conversion operator, 322–323
- deferred query execution, 307–309
- expression trees, 324–327
- extension methods, 304–306
- filtering, 311–314
- generation operators, 323–324
- grouping query results, 315–316
- grouping with nested objects, 316–317
- join clauses, 317–318

Lambda expressions, 306–307

LINQ query notation, 307

.NET and, xlvii–xlviii

partitioning operations, 319–321

providers, 327–328

queries in C# and VB, 1708

set operations, 318–319

sorting, 314–315

to SQL and syndication feeds, 1647

to SQL and WCF, 1458–1459

to SQL and XML, 1007–1008

standard query operators, 309–311

using List<T>, 298–304

LINQ to Entities, 1678–1679

LINQ to SQL

adding LINQ to SQL class, 898–899

calling products table (Northwind), 897

column names, 910–911

custom object, creating, 908

DataContext object, 903–906

DataContext object, creating, 911–912

filtering with query expressions, 916

grouping items with queries, 918–919

joins, performing with queries, 916–918

limiting called columns with query, 910

mapping database objects to LINQ objects, 902–907

O/R Designer, 899, 912–914

overview, 895–896

products object, creating, 900–902

query expressions, 914–915

querying with custom object and LINQ, 908–910

stored procedures and, 919–920

Table<TEntity> object, 907

using with LINQ to XML, 981–983

working without O/R Designer, 907–912

LINQ to XML

.NET 3.5 and, 968

overview, 967–968

querying XML documents, 974–978

reading from XML documents, 978–979

using with LINQ to SQL, 981–983

writing to XML documents, 979–981

XAttribute object, 974

XComment object, 973

XDocument object, 969

XElement object, 969–970

- XML objects from .NET 3.5, 969–973
 - XNamespace object, 971–972
 - ListBox control (Windows Forms), 1029–1031**
 - ListBox style, 1209–1210**
 - ListBox, styling (WPF), 1192–1197**
 - ListControl class (Windows Forms), 1029–1031**
 - ListenActivity (WF), 1493–1494**
 - listener ports (WCF example), 1461**
 - listeners**
 - event log listeners, 526–527
 - tracing and, 510, 512–515
 - lists, 250–266**
 - adding items dynamically, 1213–1214
 - basics, 250–252
 - binding to database (ADO.NET), 1296–1297
 - interfaces, 249
 - linked lists, 268–275
 - list binding, 1208–1211
 - read-only collections, 261
 - sorted lists, 275–278
 - lists, creating**
 - basics, 252–253
 - collection initializers, 253
 - elements, accessing, 254–256
 - elements, adding, 253–254
 - elements, inserting, 254
 - elements, removing, 256–257
 - searching, 257–258
 - sorting, 258–260
 - type conversion, 260–261
 - List<T> class. See also lists, creating**
 - access to collection classes and, 255
 - big-O notation, 295
 - collections and, 250–253
 - generics and, 224, 225
 - LINQ query using, 298–304
 - ListView control**
 - Web server controls, 1288, 1303, 1305–1307
 - Windows Forms, 1033–1035
 - local variables, scope clashes for, 32–34**
 - localization, 651–678. See also globalization**
 - with ASP.NET, 667–669
 - cultures, creating custom, 677–678
 - custom resource reader, 673–677
 - defined, 639
 - resource dictionary, 669, 671–673
 - using resources for. See resources
 - Visual Studio 2008 for. See Visual Studio 2008 for localization
 - with WPF, 669–673
 - with XAML, 671–673
 - lock statement**
 - Interlocked class and, 556
 - Monitor class and, 557
 - thread safety and, 550–555
 - logging, event. See event logging**
 - logical layer (ADO.NET Entity Framework), 1657–1658**
 - login**
 - implementing system, 1334–1336
 - Web server controls, 1336
 - logo, RSS, 1643–1644**
 - long value type, 35**
 - LongLength property, array class, 127**
 - Lookup<TKey, TElement> class, 283–284**
 - loops**
 - basics, 45–49
 - in .Net languages, 1695–1696
 - loosely coupled events (LCE), 1530–1531**
 - lower level protocols (network communications)**
 - basics, 1443–1444
 - classes, 1444
 - Smtplib object, 1444–1445
 - Socket class, 1449–1452
 - TCP classes, 1445–1446
 - TCP vs. UDP, 1448
 - TcpSend/TcpReceive examples, 1446–1448
 - UDP class, 1448–1449
 - LuridTheme theme, 1342**
- ## M
- machine configuration files, 496**
 - machine policy levels (security), 621–622**
 - main function (Windows Services), 717–718, 727–728**
 - Main() method**
 - in C#, 55
 - example of, 26
 - multiple, 55–56
 - passing arguments to, 56–57
 - making code, defined, 427**
 - Managed AddIn Framework (MAF). See add-ins**
 - managed code, defined, 4**

managed heaps

- array initialization and, 122
- defined, 8, 35
- garbage collection and, 333–334
- memory and, 331–333
- reference types and, 124

managed types, defined, 343

manifests, assembly, 472

mapping layer (ADO.NET Entity Framework), 1660

Mapping Specification Language (MSL), 1660

maps. See dictionaries

markup extensions

- binding and (XAML), 1206
- WPF and, 1154

marshaling, 755–756

MaskedTextBox control (Windows Forms), 1036–1037

.master extension (ASP.NET), 1323

master pages (ASP.NET)

- accessing content from Web pages, 1325–1326
- basics, 1311, 1323–1325
- MasterPage.FindControl() method, 1326
- MasterPage.master, 1326–1327, 1330
- nested, 1326
- in PCSDemoSite, 1326–1328

matches

- locations (WF), 1519
- Matches() static method (Regex class), 216
- in regular expressions, 216–217, 220–222

MathOperations class example (delegates), 182–184

MathTest example (invoking methods), 79–80

MD5 (Message Digest Algorithm 5), 593

MDI (Multiple Document Interface) applications, 1047–1048

members, class, 76

membership conditions (code groups), 613–614

MemberwiseClone() method, 98

memory

- blittable data types and, 990
- clean-up (C#), xlix
- memory type safe code, 14
- .NET and COM, 750

memory management

- automatic handling of, 329
- direct access to. See pointers
- garbage collection, 333–334
- reference data types, 331–333
- unmanaged resources, freeing. See unmanaged resources, freeing
- unsafe code, 339
- value data types, 330–331

menus, ribbon (VSTO), 1401–1404

MenuStrip control (Windows Forms), 1041–1042

Merge Module installer Project, 448

meshes, defined (P2P), 1628

message queues, 1555–1585

- administrative tools, 1560–1562
- architecture of, 1559–1560
- Course Order application. See Course Order application example
- creating, 1560–1563
- features of, 1557–1558
- finding, 1563
- installation, 1585
- opening, 1563–1565
- overview, 1555–1556
- products for, 1558
- queuing with WCF. See message queuing with WCF
- receiving messages, 1567–1569, 1573–1576, 1581–1583
- receiving results, 1576–1577
- sending messages, 1565–1567, 1583–1585
- transactional queues, 1577–1578
- types of, 1559
- when to use, 1556–1557

message queuing with WCF, 1579–1585

- entity classes with data contract, 1579–1580
- message receiver application, 1581–1583
- message sender application, 1583–1585
- overview, 1579
- service contract, 1580–1581

messages

- contracts (WCF), 1467, 1470–1471
- custom resource, 665–666
- for event logs, 522
- receiving, 1567–1569, 1573–1576, 1581–1583

recoverable, defined, 1559

meta-characters (regular expressions), 217

metadata

assembly, 470–471
 .NET and COM, 750
 properties (WF), 1506
 type, 470–471

methods

AddInStore class, 1254
 AddInToken class, 1255
 anonymous, 190–191
 Application class, 1018–1019
 asynchronous page requests and, 1429
 BackgroundWorker class, 571
 base versions, calling, 106
 BitArray class, 289
 BitVector32, 291
 in C#, 78
 C# rules for use of, 72–73
 called by generic delegates, 238–240
 calling outside workflow (WF), 1492
 calling Web methods (ASP.NET Ajax),
 1381–1383
 collections interfaces, 248–249
 connecting menu choices to code and,
 1090–1091
 ContextUtil class, 1540
 ContractHandle class, 1259
 DataContext object, 905–906
 DataTable, 1088–1089
 declaring, 78
 defined, 77
 entity classes, 1541–1542
 EventLog class, 520
 extension, 99–100, 304–306
 for file system management, 794
 generic, 236–238
 handler (Windows Services), 728–729
 HashSet<T> collection class, 286–287
 hiding, 104–106
 ICollection interface, 132
 IDispatch interface, 751–752
 IEnlistmentNotification interface, 704
 IEnumerator interface, 134
 IList interface, 132–133
 Interlocked class, 555–556
 invoking, 79–80

IUnknown interface, 751
 LinkedList<T> class, 269–270
 Lookup <TKey, TElement >, 283–284
 method binding, 752
 method overloading in C#, 82–83
 method parameters and return types,
 1687–1688
 ObjectContext class, 1664–1665
 ObjectQuery<T> Query Builder, 1673
 overriding for user input response, 1138–1140
 passing parameters to, 80–82
 <provider>Command classes, 856–860
 Queue and Queue<T> classes, 262–263
 ReaderWriterLockSlim, 565
 reading/writing to files, 807
 RegistryKey class, 833
 ResourceManager class, 676
 sealed, 107
 Select (XPathNavigator), 938
 ServiceController class, 743
 ServicedComponent class (Enterprise
 Services), 1531
 SortedList<TKey, TValue> class, 277
 Stack and Stack<T> classes, 267
 static, and Path class, 795
 Storyboard class, 1235–1236
 stream objects and, 794
 String class, 1212
 StringBuilder, 208
 StringInfo class, 640
 System.Drawing.Graphics, 1118–1119
 System.Object, 97–98
 System.string class, 204–205
 Table<TEntity> object, 907
 Transaction class, 686–687
 Type class, 366–367
 user controls and, 1313
 user-defined aggregates and, 996
 using generic delegate types, 240–241
 virtual, 104
 WaitHandle class, 558
 Web methods, 1381–1383
 XPathNavigator, 937–938
 XPathNodeIterator, 939
 yield statements and, 135

MEX (Metadata Exchange) endpoints, 1458, 1553

MFC containers, 783

Microsoft

- Active Directory. See Active Directory; directory services
- Foundation Class library, 1075
- Intermediate Language (IL). See Intermediate Language (IL)
- Message Queue (MSMQ) Server Core, 1558
- Microsoft.Office.Tools.Excel namespace, 1392–1394
- Microsoft.Office.Tools.Word namespace, 1392–1393
- Microsoft.SqlServer.Server namespace classes, 987–988
- Microsoft.Win32 namespace, 791, 830
- Windows history, xliii–xliv
- Windows Peer-to-Peer Networking platform, 1625, 1629–1632

MMC (Microsoft Management Console)

snap-ins

- .NET Framework Configuration tool, 501
- Services snap-in, 735–736

modifier keywords, 112–114

modules, creating, 473–474

Money example (ToString() method), 98–100

Monitor class (synchronization), 557

MoveTo() method (files), 794, 800–801

moving files, 800–805

MSMQ (Microsoft Message Queue) Server Core, 1558

MTA (multithreaded apartment), 753–754

multi-targeting .NET Framework, 436–438

multicasting

- messaging queuing support, 1558
- multicast delegates, 187–190
- MulticastIteration example, 189–190

multidimensional arrays, 125

multimaster replication (Active Directory), 1588

multiple casting, 171–175

multiple catch blocks, 382–385

multiple inheritance, 102

multithreading

- with ambient transactions, 697–701
- multithreaded apartments (MTAs), 569–570, 753–754

Mutex class (synchronization), 558, 559–560

myArray variable, 122

N

naked type constraints, defined, 233

Name properties (Windows Forms elements), 660

named colors (GDI+), 1115

names. See also naming conventions;

peer names

- casing of, 68–69
- distinguished (Active Directory), 1599–1601
- relative distinguished names (Active Directory), 1599
- server name (Active Directory), 1599
- styles of, 69
- user class properties, 1613–1614
- user principal name (Active Directory), 1601

namespaces

- ADO.NET Entity Framework, 1656
- aliases, 54–55
- assemblies and, 472
- basics, 52–55
- defined, 27, 52
- defining (Ajax Library), 1376
- defining with .Net languages, 1682–1683
- GDI+, 1095
- names of, 70
- .NET, 18–19
- for .NET data access, 846
- using directive, 53–54
- VSTO, 1391–1392
- WPF, 1156–1159
- XNamespace object, 971–972

naming conventions

- basics, 68–72
- constraints, 893
- database tables, 892–893
- default naming context (Active Directory), 1600–1601
- for generic types, 226
- stored procedures, 893
- threads, 540
- trace sources, 511

native image generator, 494–495

Native Image Service, 495

NativeMethods class, 711

navigating Web sites (ASP.NET), 1311, 1328–1331

nested master pages (ASP.NET), 1326

nested objects, grouping queries with, 316–317**nested partial classes, 96****nested scopes, 695–697****nested try blocks, 387–389****.NET and COM. See also COM (Component Object Model)**

client application, creating with sink object, 781–782

COM callable wrapper (CCW), 771

COM client, creating, 778–779

COM client, using .NET component from, 771

COM component, creating, 756–762

COM Interop attributes, 774–777

COM registration, 777–778

connection points, adding, 780–781

error handling, 754

event handling, 754–755

fundamentals, 750

interfaces, 751–752

marshaling, 755–756

memory, 750–751

metadata, 750

method binding, 752

.NET component, 771–772

registration, 753

runtime callable wrapper (RCW), 762–764

threading, 753–754

type library, creating, 772–774

.NET applications

<codeBase> element, 497–498

configuration categories, 496

configuring directories for assembly searches, 497–499

hardware and software requirements, 445

<probing> element, 498–499

.NET architecture

ASP.NET applications, creating, 19–20

C# relationship to .NET, 4

CLR, 4

COM/COM+, 6

enterprise applications and C#, 22–23

language interoperability, 5

Microsoft Intermediate Language (IL). See Intermediate Language (IL)

namespaces, 18–19

.NET Framework classes, 17–18

overview, 3

performance improvement, 4–5

platform independence, 4

Visual Basic 2008, 5–6

Visual C++ 2008, 6

Windows Communication Foundation (WCF), 22

Windows Controls, 21

Windows Forms, creating, 21

Windows Presentation Foundation (WPF), 21

Windows Services, 21–22

.NET Framework

advantages of, xlviv–xlv

base classes, 28

basics, xlvi

C# and, xlviii

classes, 17–18

classes and file management, 792–794

code group membership conditions, 613–614

Configuration tool, 501, 505

Enterprise Services, 684–685, 1528

globalization/localization and, 639

multi-targeting with VS 2008, 436–438

.NET 3.5, xlv–xlvii

.NET runtime (CLR). See CLR (Common Language Runtime)

programming with SQL Server. See SQL Server registry and, 828, 830–833

significance of, xliii

W3C standards supported in, 922

XML and, 921

net.exe command-line utility (Windows Services), 734–736**neutral cultures, 641–642****new keyword, 76, 105, 120****new operator (arrays), 122****news feeds, 1643–1644****Ngen.exe (native image generator), 494–495****Nikhil's Web Development Helper, 1381****no-touch deployment (NTD), 457–458****nodes, inserting with XmlDocument class, 934–936****nonblittable data types, 756****nonrepeatable reads (transactions), 701****Not() method, 290****null coalescing operator (??), 147****nullable types**

basics, 243–244

operators and, 147

numbers

- rounding, 168
- truncating, 168
- version, 499–500

numeric types, 209–210

numerical conversions. See SimpleCurrency example (user-defined casts)

O

O/R Designer

- basics, 899
- and custom objects, 912–914

objects

- Active Directory, 1589, 1621–1623
- Application objects basics, 1308
- C# object-orientation, 25–26
- comparing for equality, 153–154
- defining instances with XAML, 1206–1207
- keyword, 103
- LINQ to objects provider, 327
- mapping database to LINQ objects, 902–907
- native ADSI objects, accessing, 1607–1608
- new objects, creating (Active Directory), 1605–1606
- Object Browser window (VS 2008), 425–426
- Object class, 97, 223, 231
- object collections (Active Directory), 1603–1604
- object initializers, xlvii, 1691
- object-type collections, 247–248
- ObjectContext class, 1664–1666
- performance objects, 527
- pooling (Enterprise Services), 1530
- queries, 1670–1674
- simple binding CLR objects, 1203–1206
- strongly typed, 1588
- support in .NET, 7–8
- tracking, 1674–1675
- types, 40, 223, 230
- used in data binding, 1076–1080
- user objects, searching for (Active Directory), 1611–1616

Office 2007 object model, 1391

Office applications, 1385. See also Visual Studio Tools for Office (VSTO)

OfType() extension method, 312

OLE DB provider, 1724–1727

one-time binding, 1200

one-way binding, 1200

OnPaint() method

- CapsEditor example, 1135–1137
- painting shapes with, 1099–1100

OnTimer() method, 530

OOP extensions, JavaScript, 1375–1378

open unlocked state (WF), 1509–1511

OpenFileDialog class, 1226

operating systems and .NET-based applications, 445

[OperationBehavior] attribute (WCF), 1475–1476

OperationContract properties (WCF), 1470

operators

- aggregate, 321–322
- available for overloading, 163–164
- conversion, 322–323
- defined, 78
- generation, 323–324
- pointer member access, 346
- standard query (LINQ), 309–311
- using with pointers, 341, 344–346

operators, C#

- checked and unchecked, 145
- comparison operator, 154
- compiler processing of, 156–157
- conditional, 144–145
- is operator, 146
- null coalescing operator (??), 147
- nullable types and, 147
- as operator, 146
- operator overloading. See overloading, operator
- overview, 141–143
- precedence of, 147–148
- shortcuts, 143–144
- sizeof operator, 146
- typeof operator, 146

optimization of code, 427–428

out parameters, 82

output parameters, 861–863

outsourcing translations, 666–667

OverflowException class, 379

overloading, method, 82–83

overloading, operator

- comparison operator, 161–163
- compiler processing and, 156–157
- fundamentals, 156–157

operators available for, 163–164
 Vector struct example, 157–163

overriding

defined, 40
 override keyword, 104, 120
 publisher policies, 505–506

P

P2P. See peer-to-peer (P2P) networking

page coordinates (GDI+), 1113–1114

Page Setup, printing (GDI+), 1142

Page_Load() event handler

ASP.NET code model, 1279, 1313
 calendar controls, customizing, 1297, 1299
 server controls (ASP.NET), 1292

Page_PreInit() event handler (ASP.NET), 1343

PageRequestManager class (Ajax Library), 1378–1380

painting shapes (GDI+), 1099–1100

Panel control (Windows Forms), 1037

panes (VSTO project types), 1388–1390, 1404–1405

ParallelActivity (WF), 1491–1492

parameters

Activate() method, 1255–1256
 adding constructors with, 110–112
 AttributeUsage attribute, 360–361
 binding to activities, 1513–1514
 methods using generic delegates as, 240–241
 modifiers, 1688–1689
 out, 82
 parameter type contra-variance, 194
 ParameterizedThreadStart delegate, 540–541
 passed to constructor of persistence service, 1518
 passing to methods, 80–82
 passing to workflows, 1511–1512
 ref, 81–82
 returning to workflows, 1511–1512
 specifying custom attribute, 361–362
 stored procedures returning output, 861–863

partial classes/keyword, 95–96

partial-page postbacks, 1355, 1368–1369

partitioning operators, 310, 319–321

Pascal casing, 68–69

Passport Authentication (ASP.NET), 1331

passwords, resetting (Active Directory), 1618–1619

paths

for opening message queues, 1564
 Path class and file management, 794–795
 PATH index, 1011

pattern search strings (regular expressions), 216–217

PCSDemoSite

master pages in, 1326–1328
 navigating, 1330–1331
 security, 1336–1339
 themes in, 1340–1344

PCSLibraryDemo project, 1374

peer names

defined, 1629
 defining, 1629
 Peer Name Resolution Protocol (PNRP), 1625, 1629–1631
 registering, 1633–1634
 resolving, 1634–1635

peer-to-peer (P2P) networking, 1625–1642

applications, building, 1632–1642
 client-server architecture, 1626
 forums, lii
 Microsoft Windows platform, 1625, 1629–1632
 overview, 1625–1629
 P2P architecture, 1627–1628
 Peer Name Resolution Protocol (PNRP), 1625, 1629–1631
 PeerCollaboration class, 1639
 People Near Me (PNM), 1625, 1632
 System.Net.PeerToPeer namespace. See System.Net.PeerToPeer namespace
 System.Net.PeerToPeer.Collaboration namespace. See System.Net.PeerToPeer.Collaboration namespace
 terminology, 1628–1629

pens (GDI+), 1116–1118

People Near Me (PNM), 1625, 1632

perfmon.exe, 531–532

performance

collections, 294–296
 generics and, 224–225
 pointers and, 339–340, 352–355

performance monitoring

- basics, 527
- classes, 528
- perfmon.exe, 531–532
- Performance Counter Builder, 528–529
- performance counts, 509
- Performance Monitor, 531–532, 745
- PerformanceCounter components, 529–531

permissions

- asserting, 611–613
- changing code group, 627–628
- code access security and, 618–621
- declarative, 606
- defined, 602
- demanding programmatically, 605–606
- denying, 610–611
- implicit, 610
- managing, 626
- provided by CLR, 603–605
- requesting, 606–610
- sets, creating/applying, 628–630
- sets, defined, 602
- SQL Server levels, 986–987

persistence service (WF), 1514, 1517–1518

persisting DataSet changes, 884–888

phantom reads (transactions), 701

phases, transaction, 680–681

PictureBox control (Windows Forms), 1035

pinning windows, 426

pipeline architecture (add-ins), 1253–1254

platform independence (IL), 4

platform invoke services, 783–787

Playground example (regular expressions), 216–220

plug-ins. See add-ins

PNM (People Near Me), 1625, 1632

PNRP (Peer Name Resolution Protocol), 1625, 1629–1631

Point/PointF structs (GDI+), 1102–1103

pointer member access operators, 346

pointers, 339–355

- advantages/disadvantages of, 340
- arithmetic of, 344–345, 349–350
- casting between pointer types, 343–344
- casting to integer types, 343
- to class members, 346–347
- optimizing performance with, 352–355
- PointerPlayaround example, 347–352

- reasons for using, 339–340

- sizeof operator, 345

- to structs, 345–346

- syntax, 341–343

- unsafe keyword, 340–341

- void pointers, 344

poison queues, defined, 1557

policies, security

- levels, 621–622

- managing, 622–626

polling (asynchronous delegates), 535–536

pooling, object (Enterprise Services), 1530

pools, thread, 543–545

pop-up menu for rows

- attributes, 1089–1090

- code, 1085–1088

- dispatching methods, 1090–1091

- manufactured tables and rows, 1088–1089

- selected rows, 1091–1092

populating data sets

- with data adapters, 883–884

- from XML, 884

ports

- listener (WCF example), 1461

- number (Active Directory), 1599

- types (WSDL), 1458

postbacks

- Ajax and, 1357

- asynchronous, 1355

- defined, 1275

- partial-page, 1368–1369

postfixes, expression, 144

power events (Windows Services), 746–747

#pragma preprocessor directives, 66

prefixes, expression, 144

preparing transaction phase, 680

preprocessor directives, 63–66

presentation service layer, 1528

previewing, print (GDI+), 1142–1146

primary interop assembly (COM), 764

primary keys, setting, 874–875

principal classes, 583–585

printable characters, defined, 815

printing

- and previewing in GDI+, 1141–1146

- WebBrowser control for, 1437

priority nodes, 270, 273–274

Priority property (Message class), 1571–1572

- PriorityDocumentManager class, 272**
- private assemblies, 16, 472–473**
- private keys, strong names and, 486–487**
- private queues, defined, 1559**
- <probing> element, 496, 498–499**
- products object, creating (Northwind), 900–902**
- products table, calling (Northwind), 897**
- Professional ASP.NET 2.0 (Wiley), 1308**
- Professional ASP.NET 3.5: in C# and VB (Wrox Press), 216**
- programming**
 - Active Directory. See Active Directory, programming
 - object-oriented, xlviv, xlviii
- programming guidelines (C#)**
 - fields, 73
 - identifiers, rules for, 66–67
 - naming conventions, 68–72
 - properties/methods, rules for using, 72–73
 - usage conventions, 67–68
- ProgressBar control (Windows Forms), 1035–1036**
- ProjectInstaller class, 730–731**
- projection operators, 310**
- projects**
 - deployment, 444
 - vs. solutions, 412–415
- projects, Visual Studio 2008**
 - creating new, 406–407
 - selecting type, 407–410
 - startup, setting, 415
- projects, VSTO**
 - features, 1389–1390
 - structure of, 1394–1396
 - types, 1386–1389
- promotable transactions, 690–692**
- properties**
 - access modifiers for, 84–85
 - accessing by name (Active Directory), 1603
 - ACID (transactions), 681
 - AddInToken class, 1255
 - appearance (Control class), 1024
 - Application class, 1018–1019
 - application setup (Solution Explorer), 450–452
 - array class, 127
 - attached (WF), 1506–1508
 - auto-implemented, xlviv, 85
 - BackgroundWorker class, 571–572
 - Binding class, 1202–1203
 - BitVector32, 291
 - C#, rules for use of, 72–73
 - ChannelFactory class, 1483
 - as child elements (XAML), 1152
 - collections interfaces, 248–249
 - Custom Actions editor, 464–465
 - DataColumn object, 868
 - DataContext object, 905–906
 - DataMember properties (WCF), 1468
 - defined, 77
 - defining in .NET languages, 1690–1691
 - deployment projects, 453–454
 - DllImport, 784–785
 - DoubleAnimation, 1232
 - entity classes, 1541, 1542
 - EventLog class, 520
 - EventSourceCreationData, 521–522
 - EventTrigger class, 1236
 - FileInfo/DirectoryInfo class, 793–794
 - fundamentals, 83–84
 - header properties, 1427–1428
 - ICollection interface, 132
 - IEnumerator interface, 134
 - IList interface, 132–133
 - to limit searches (Active Directory), 1609–1610
 - LinkedList <T> class, 269–270
 - LinkedListNode class, 269
 - Lookup<TKey, TValue>, 283–284
 - message queuing, 1561–1562
 - Name/Text (Windows Forms elements), 660
 - names, user class (Active Directory), 1613–1614
 - ObjectContext class, 1664–1665
 - ObjectDataProvider class, 1207
 - OperationBehavior (WCF), 1475–1476
 - OperationContract properties (WCF), 1470
 - Path class, 795
 - Properties window, VS 2008, 422–424
 - PROPERTY index, 1011
 - property triggers (timeline), 1229–1233
 - PropertyManager object, 1078–1080
 - Queue and Queue<T> classes, 262–263
 - read-only/write-only, 84
 - ReaderWriterLockSlim, 565
 - Rectangle struct, 1105
 - RegistryKey class, 832

properties (continued)

- search limits and (Active Directory), 1609–1610
- ServiceContract (WCF), 1469
- ServiceController class, 738–739
- services, 725–726, 732–733
- size/location (Control class), 1023–1024
- SortedList<TKey, TValue> class, 277
- [SqlUserDefinedType] attribute, 990
- Stack and Stack<T> classes, 267
- stream objects and, 794
- StreamReader class, 817
- Syndication Reader, 1646
- System.Exception, 385
- Timeline, 1232
- Transaction class, 686–687
- Trigger class, 1184
- Type class, 365–366
- user controls and, 1313
- validation controls and, 1289
- virtual, 104
- Web application installer, 455–456
 - as XML attributes, 1152
- XPathNodeIterator, 939
- yield statements and, 135

protected internal modifier, 112, 114

protocols, Active Directory binding, 1598–1599

providers

- ADSI, 1598–1599
- LINQ, 327–328
- <provider>Command classes, 856–860

proxies

- creating for clients (WCF), 1482–1483
- HTTP/FTP requests and, 1428–1429

public keys, strong names and, 486–487

public modifier, 112, 114

public queues, defined, 1559

publisher policy files, 496, 504–506

publishing

- applications, 459
- Web sites, 444, 447–448

Q

quantifier operators, 310

queries. See also LINQ (Language Integrated Query)

- object, 1670–1674

- Query Builder methods, 1673–1674
- query expressions, 914–916
- QueryInterface() method, 752

querying SQL Server database (LINQ)

- custom objects, 908–910
- grouping items with queries, 918–919
- limiting called columns, 910
- outputting XML, 982–983
- performing joins, 916–918
- query expressions, 914–915

querying XML documents, 974–978

queues

- basics, 261–266
- configuring with Component Services Explorer, 1537
- queued components (Enterprise Services), 1530
- Stack<T> big-O notation, 295
- workflows (WF), 1492

QuickArray example (pointers), 354–355

R

race conditions (threading), 545–548

Racer class (lists), 251–261

RadialGradientBrush (WPF), 1164–1165

radio button control (Windows Forms), 1028

Range() method, 324

Rank property (Array class), 127

RCW (runtime callable wrapper), 762–764

read methods (XmlReader class), 925–927

read-only collections, 261

read-only fields/keyword, 90–91

read-only properties, 84

ReaderWriterLockSlim class (synchronization), 564–568

reading

- ACLs from files/directories, 824–825
- to files, 805–807
- to files with FileStream class, 810–812
- to isolated storage, 839–844
- to the registry. See registry
- to text files, 815–821

ReadWriteText example, 819–821

Really Simple Syndication (RSS), 1643

receiving messages, 1567–1569, 1573–1576, 1581–1583

records

- deleting, 861
- inserting, 862–863
- updating, 860–861

recoverable messages, defined, 1559**Recoverable property (Message class), 1572****Rectangle/RectangleF structs (GDI+), 1105****red-green-blue color values, 1114–1115****ref parameters, 81–82****refactoring applications, 434–436****reference types. See also arrays**

- basics, 35–36, 40–42
- comparing for equality, 153–154
- declaring arrays of, 123–124
- declaring with .NET languages, 1683–1684
- generics and, 224
- reference data types, 331–333
- and value types, 8

references

- JavaScript client-side, 1369
- ReferenceEquals() method, 97, 153
- referents, defined, 339
- service (WCF), 1464–1465
- to shared assemblies, 493–494
- Web service, 1369

reflection

- assembly class, 370–371
- custom attributes. See custom attributes
- defined, 17, 357
- System.Type class. See System.Type class

Reflector software tool, 853, 1500**regasm utility, 778****regedit, 829–830****Region class (GDI+), 1106****#region preprocessor directives, 65****regions**

- date formatting and, 644–645
- number formatting and, 643–644

registration

- COM, 777–778
- .NET and COM, 753

registry

- basics, 828
- .NET Registry classes, 830–833
- overview, 829–830
- Registry editor (VS 2008), 461
- registry hives, 829–830

Registry Search (Launch Conditions editor), 465

RegistryKey class, 830–834

SelfPlacingWindow example, 833–839

regular expressions

- asterisk (*) as quantifier in, 218
- backslash (\) in, 217
- Beginning Regular Expressions* (Wrox Press), 215
- escape codes in, 215
- escape sequences, 217–218
- fundamentals, 214–216
- input strings, 216
- matches/groups/captures, 220–222
- Matches() static method (Regex class), 216
- meta-characters, 217
- parentheses () in, 220
- pattern search strings, 216–217
- RegexOptions enumerations, 217
- RegularExpressionsPlayground example, 216–220
- special characters in, 218
- square brackets ([]) in, 218

relational data

- converting to XML, 952–954
- XML data type and, 1003

relationships

- ADO.NET Entity Framework and, 1666–1670
- data, 873–874

relative distinguished names (Active Directory), 1599**release and debug builds, 427****Release configuration (Visual Studio), 429****Replace() method (System.String), 206****replication (Active Directory), 1588, 1591–1592****report queues, defined, 1559****requests. See also WebRequest/WebResponse classes**

- asynchronous page, 1429
- TCP network, 1444

Resgen.exe, 651–652**resource files**

- event logging, 522–526
- localization and, 651, 653–658
- Resource File Generator, 651–652
- ResourceManager, 654–659

resource readers, custom

- application example, 677
- basics, 672
- DatabaseResourceManager class, 676–677
- DatabaseResourceReader class, 674–675, 677
- DatabaseResourceSet class, 676

resources

- automatic fallback for, 666
- custom resource messages, 665–666
- management in .Net languages, 1701–1703
- managers, custom, 703–709
- resource dictionary (XAML), 669, 671–673
- ResourceWriter, 652–653
- System.Resources namespace, 659
- transactional, 704–709
- WPF, 1178–1181

response queues, 1559, 1577

resultLabel control Text property, 1293–1294

return statements, 49

return type covariance, 193

RevokeLifetimeToken() method, 1258–1259

ribbon menus (VSTO), 1390, 1401–1404

RichTextBox control (Windows Forms), 1036–1037

Roeder, Lutz, 1500, 4500

role-based security

- defined, 12
- Enterprise Services and, 1530
- fundamentals, 585

Rooms table (MeetingRoomBooker example), 1296

RotateTransform element (WPF), 1162

round-robin scheduling (threads), 543

rounding numbers, 168

rows. *See also* pop-up menu for rows

- data, 868–871
- deleting in data tables, 887
- filtering, 1065–1067
- filtering by data, 1066
- filtering on state, 1066–1067
- inserting in data tables, 885–886
- sorting, 1068
- updating existing (data tables), 887

RSS (Really Simple Syndication), 1643–1645

rules, custom validation, 1223–1224

runat="server" attribute, 1275

runtime. *See also* CLR (Common Language Runtime)

- callable wrapper (RCW), 762–764
- schema generation, 871
- settings, configuring, 496, 503
- versions, 506–507
- workflow runtime (WF), 1514–1515

S

SafeHandle type, 710

safety palette (GDI+), 1116

safety permission levels (SQL Server), 986–987

SampleClientSetupSolution example, 460

sans-serif font families, 1126

satellite assemblies

- defined, 473
- for languages, 663
- localization with XAML, 671, 673

saving

- files with Unicode characters, 651
- SaveFile() method, 819–820
- SaveSettings() method, 840–842

ScaleTransform element (WPF), 1162

scaling, client-server architecture, 1626

sc.exe command-line utility (Windows Services), 734–737

schemas. *See also* XML schemas

- Active Directory, 1589, 1592–1593
- defined, 866
- generation (data tables), 871–873
- naming context (Active Directory), 1612–1613

SCM (Service Control Manager), 717

scopes. *See also* variable scope

- clashes, 32–34
- nested (transactions), 695–697
- rules for determining, 31

<script> tags (ASP.NET Web Forms), 1275

ScriptManager control (ASP.NET Ajax), 1359, 1368–1370

scrollable windows, drawing, 1107–1113

scrolling, 1079–1080

sealed classes/methods, 107

sealed keyword, 107

searching. *See also* regular expressions

- in Active Directory, 1608–1611

- Advanced Query Syntax (AQS), 1728–1729
 - with DSML, 1621–1623
 - lists, 257–258
 - in Vista, 1722–1729
- security**
 - access control to resources, 599–601
 - application deployment and, 460
 - assemblies and, xlv
 - authentication and authorization. See authentication and authorization
 - code access security. See code access security
 - configuring with Component Services Explorer, 1536
 - distributing code with certificates, 631–637
 - distributing code with strong names, 630–631
 - encryption. See encryption
 - of files, 825–828
 - managing code groups/permissions, 626
 - managing security policies, 622–626
 - .NET overview, 12
 - role-based (Enterprise Services), 1530
 - secure transfers and key exchanges, 596–599
 - SecurityAction enumeration values, 608–609
 - settings, configuring, 496
 - SQL Server and, 986–987
 - turning on and off, 626
 - WCF and, 1456
- security (ASP.NET), 1331–1339**
 - directories, securing, 1337–1339
 - forms authentication, 1331–1334
 - login system, 1334–1336
 - login Web server controls, 1336
 - PCSDemoSite, 1336–1339
 - security Setup Wizard, 1332–1334
 - Web Forms, 1276
- SELECT statement**
 - combining with XQuery, 1008–1009
 - UDTs and, 988, 994–995, 997
- SelfPlacingWindow example (registry), 833–839**
- Semaphore class (synchronization), 558, 560–562**
- Send() method (queues), 1565**
- sending messages, 1565–1567, 1570–1572, 1583–1585**
- sentinel values, defined, 1074**
- sequential workflows (WF), 1509**
- serialization**
 - data contracts (WCF), 1468
 - serializing objects in XML, 956–965
 - in XML, 963–965
- serif font families, 1125**
- server console application, building, 1450–1451**
- server controls (ASP.NET)**
 - control palette, 1283
 - data Web server controls, 1287–1288
 - example, 1289–1294
 - overview, 1279–1283
 - standard Web server controls, 1283–1286
 - validation Web server controls, 1288–1289
- Server Explorer**
 - controlling Windows Services with, 737
 - window (VS 2008), 426
- server names (Active Directory), 1599**
- service configuration**
 - programs (Windows Services), 718
 - Service Configuration Editor (WCF), 1463, 1466
- service contracts (WCF), 1467–1470, 1580–1581**
- service façade (WCF), 1549–1553**
- Service Host (WCF example), 1461–1464**
- service-main function (Windows Services), 717–718**
- service programs (Windows Services), 716–718**
- Service Trace Viewer tool (WCF), 1467**
- ServiceBase class (Windows Services), 726–727**
- ServiceBehavior property (WCF), 1471–1472**
- [ServiceContract] attribute, 1469**
- ServiceController class**
 - application (Windows Services), 737–745
 - defined, 719
- ServiceComponent class (Enterprise Services), 1531**
- ServiceInstaller class, 731–734**
- ServiceInstallerDialog class, 734**
- ServiceModel Metadata Utility tool (Svcutil.exe), 1482–1483**

ServiceProcessInstaller class, 731–734

services, workflow (WF)

- basics, 1515–1516
- custom services, 1520–1521
- persistence services, 1517–1518
- tracking service, 1518–1520

set operations, LINQ and, 318–319

set operators, 310

Setup installer Project, 448

SetValue() method, 127–128

shallow comparisons, 162

shapes

- drawing, 1096–1099
- painting, 1099–1100
- in WPF, 1159–1162

shared assemblies

- basics, 472–473
- creating, 489
- defined, 16–17
- delayed signing of, 492–493
- global assembly cache (GAC), 487–489
- installing, 491
- native image generator, 494–495
- .NET applications, configuring. See .NET applications
- references to, 493–494
- strong names and, 486–487, 489–490
- using, 491–492
- versioning and. See versioning

shared classes in System.Data namespace, 847

SharePoint workflow templates(VSTO), 1388

shield icons (Vista), 1713–1715

shortcuts, operator, 143–144

signatures, encrypted, 594–596

SignIn()/SignOut() methods (PeerCollaboration class), 1639

signing shared assemblies, 492–493

Silverlight applications, 1249

simple controls (WPF), 1167–1168

SimpleAppSolutionSetup example (deployment), 454–455

SimpleClientApp example (deployment), 449–454

SimpleCurrency example (user-defined casts), 165–174

SimpleExceptions example (catch blocks), 382–385

SimpleWebApp example (deployment), 455–456

single-threaded apartments (STAs), 569–570, 753

sink objects, creating client applications with, 781–782

<siteMapNode> element attributes, 1329

sites, domain controller, 1590

Size/SizeF structs (GDI+), 1103–1104

sizeof operator, 146, 345

SkewTransform element (WPF), 1162

Smart Device CAB installer Project, 448

smart tags (VSTO), 1389

SmtpClient object, 1444–1445

SOAP (Simple Object Access Protocol)

- faults, 1476–1477
- WCF and, 1457

Socket class, 1449–1452

SolicitColdCall example (user-defined exceptions), 389–390

solid brushes, 1117

SolidColorBrush (WPF), 1163–1164

solutions vs. projects, 412–415

solutions, VSTO

- application-level add-ins, 1398–1400
- architecture of, 1386
- basics, 1398–1399
- document-level customization, 1398
- interacting with applications/documents, 1400–1401
- UI customization, 1401–1405

Sort() method (generic delegates), 240–241

SortedDictionary<TKey, TValue> class, 285

SortedDictionary<TKey, TValue> class, 285, 295

- arrays, 129–131
- cultures and, 650–651
- data within DataView, 1068
- LINQ, 314–315
- lists, 258–260
- operators, 310

source files, comments within, 60

sources

- event sources, 521–522, 525
- tracing architecture, 510–511

SpecialFolder enumeration (Vista), 1715

specifiers, numeric type format, 209–210

SplitContainer control (Windows Forms), 1038

SQL (Structured Query Language)

LINQ to SQL. See LINQ to SQL
 SqlCoordinate struct, 988–989, 991
 [SqlUserDefinedType] attribute, 990

SQL Server, 985–1013

key generation with, 890–892
 Microsoft.SqlServer.Server namespace
 classes, 987–988
 .NET runtime and, 985, 986–987
SQL Server 2005 Express Edition Starter Kit
 (Wiley), 985
 stored procedures and, 998–1000
 triggers, 1001–1003
 user-defined functions, 1000–1001
 user-defined types. See user-defined types
 (UDTs)
 XML data type and. See XML data type

SQL Server Management Studio

binding columns and, 1076
 XML schemas and, 1012–1013

STA (single-threaded apartment)

basics, 753
 COM objects, 771

stacks

basics, 266–268
 defined, 35
 memory and, 330–333
 pointers, defined, 330
 Stack<T> big-O notation, 295
 stack-based Arrays, creating, 352–354
 stackalloc keyword, 352–354
 StackOverflowException class, 379
 StackPanel container (WPF), 1172–1173
 and value types, 8

startup settings, configuration, 496**state machine workflows (WF), 1509–1511****state management (ASP.NET), 1274–1275****static classes/keyword, 96–97****static constructors, 87–89****static Equals() method, 154****static members**

generics and, 235
 and .NET languages, 1692–1693

static methods, Path class and, 795**static XML documents, querying, 975–976****StaticResource markup extension,
1181–1182****storage, isolated, 839–844****Store Schema Definition Language (SSDL),
1657–1658****stored procedures. See also triggers**

calling with command objects, 860–863
 LINQ to SQL and, 919–920
 naming conventions for, 893
 SQL Server and, 998–1000
 using in data adapters, 883–884

storing entity changes, 1677–1678**storyboards, 1229, 1235–1236****streams**

basics, 808–810
 BinaryFileReader example, 812–815
 buffered, 810
 FileStream class, 810–812
 streamed XML, reading/writing, 924
 StreamReader class, 809, 816–818
 StreamWriter class, 809, 818–819

strings

building, 205–208
 connection strings for adding events, 1300
 creating UDTs and, 992
 displaying, 1063–1064
 format strings, 59–60, 209–214
 FormattableVector example, 212–214
 for list binding, 1212–1213
 managing connection, 850–852
 StringArrayConverter class, 1212–1213
 StringBuilder members, 208–214
 StringCollection class, 820
 StringInfo class, 640
 System.string class methods, 204–205
 types, 40–42

strong data typing

IL and, 8–9
 language interoperability and, 9–11
 strongly typed XML, 1011–1013

strong names

creating, 489–490
 defined, 17
 distributing code with, 630–631
 integrity using, 486–487

structs

vs. classes, 76, 102
 constructors for, 94–95
 fundamentals, 92–93
 inheritance and, 94, 102
 pointers to, 345–346

structs (continued)

- System.Drawing namespace, 1102
- as value types, 93–94
- Vector struct example (operator overloading), 157–163
- Structured Query Language (SQL). See SQL (Structured Query Language)**
- styles, defining (WPF), 1177–1178**
- SubordinateTransaction class, 686**
- Sum() method, 321–322**
- super keyword, 106**
- SuppressFinalize() method, 339**
- .svc files, 1481**
- switch statement (.NET languages), 1694–1695**
- switch . . . case statements, 44–45**
- switches, trace. See tracing**
- symbols, debugger, 428**
- symmetric encryption keys, 591–594**
- symmetric key exchange, 596–599**
- synchronization, 549–568**
 - basics, 549–550
 - events and, 562–564
 - Interlocked class, 555–556
 - lock statement, 550–555
 - Monitor class, 557
 - Mutex class, 559–560
 - ReaderWriterLockSlim class, 564–568
 - Semaphore class, 560–562
 - WaitHandle class, 557–558
- synchronous programming, 1555**
- SyncRoot pattern, 552, 554**
- syndication**
 - Really Simple Syndication (RSS), 1643
 - syndication feeds, 1647–1651
 - Syndication Reader application, 1644–1647
 - Syndication Service Library template, 1647
 - System.ServiceModel.Syndication namespace, 1643–1644
- Sys.Debug class, 1381**
- system access-control list (SACL), 599**
- system queues, defined, 1559**
- system resources, WPF and, 1180**
- System.AddIn assembly. See add-ins**
- System.ApplicationException class, 378**
- System.Convert object, 168**
- System.Data namespace, 847**
- System.Data.Common namespace, 847**
- System.Data.OleDb namespace, 749, 1725**
- System.DirectoryServices namespace, 749, 1597–1598**
- System.DirectoryServices.Protocols namespace, 1587, 1596–1597, 1621**
- System.Drawing namespace, 1095, 1102**
- System.Drawing.Brush class, 1117**
- System.Drawing.Font class, 1126**
- System.Drawing.Graphics class, 1118–1119**
- System.Drawing.Pen class, 1117–1118**
- System.EnterpriseServices, 684–685**
- System.Exception properties, 386**
- System.GC class, 334**
- System.Globalization namespace. See globalization**
- System.IO namespace, 378, 791**
- System.Linq namespace, 305**
- System.Messaging namespace, 1555**
- System.Net namespace, 3**
- System.Net.PeerToPeer namespace**
 - code access security, 1635
 - example application, 1636–1638
 - registering peer names, 1633–1634
 - resolving peer names, 1634–1635
- System.Net.PeerToPeer.Collaboration namespace**
 - detecting peer presence, 1639–1640
 - example application, 1640–1642
 - managing contacts, 1639–1640
 - overview, 1638–1639
 - peers, discovering, 1639
 - signing in/out, 1639
- System.Net.Sockets namespace, 3**
- System.Net.WebClient class, 4**
- System.Object base class, 108–109, 112**
- System.Object methods, 97–98**
- System.Object.ValueType, 94**
- System.Reflection namespace, 370, 476**
- System.Resources namespace (localization), 639, 659**
- System.Runtime.InteropServices namespace, 774–775**
- System.Security.AccessControl namespace, 599**
- System.Security.Cryptography namespace, 592**
- System.Security.Principal namespace, 584**

System.ServiceModel.Syndication
 namespace, 1643–1644

System.ServiceProcess namespace,
 718–719

System.String class methods, 204–205

System.SystemException class, 378

System.Text.RegularExpressions, 203

System.Text.RegularExpressions
 namespace, 215

System.Text.StringBuilder class, 206

System.Threading namespace, 584

System.Threading.Timer class, 568

System.Transactions namespace, 685–687

System.Type class
 properties and methods, 365–367
 TypeView example, 367–370

System.ValueType, 102

System.Web.Extensions.Design.dll, 1359

System.Web.Extensions.dll, 1359

System.Web.UI.WebControls.WebControl, 1319

System.Windows namespace, 1154

System.Windows.Forms namespace, 1023,
 1025, 1076

System.Windows.Forms.Control class, 21

System.Windows.Media.Animation
 namespace, 1229

System.Windows.Shapes namespace, 1161

System.Workflow.Activities namespace, 1489

System.Xml classes, 923–924

System.Xml namespace, 922–923

System.Xml.Linq namespace, 327, 969

System.Xml.Serialization namespace,
 956–957

System.Xml.XPath namespace, 937–942

System.Xml.Xsl namespace, 942–947

T

T-SQL stored procedures, 998–1000

TabControl/TabPage controls (Windows Forms),
 1038–1039

tables
 creating with XML data, 1003–1005
 manufactured, 1088–1089
 Table per Hierarchy (TPH), 1666–1668
 Table per Type (TPT), 1668–1670
 TableLayoutPanel container (Windows Forms),
 1037–1038

Table<TEntity> object, 907
 temporary, accessing with triggers, 1002

tabular data, displaying, 1062–1063. See also
DataGridView control

tags for XML documentation, 61

TakesAWhileCompleted() method, 537

/target switches, 57

task dialogs (Vista), 1718–1721

task panes (VSTO), 1404–1405

TCP (Transmission Control Protocol)
 classes, 1445–1446
 network requests, 1444
 TcpClient example (Windows Services),
 722–725
 TcpSend/Receive examples, 1446–1449
 vs. UDP, 1448

templates
 for data display, 1303–1307
 data templates and WPF binding,
 1214–1217
 FrameworkTemplate class (WPF), 1184–1192
 InfoPath form templates (VSTO), 1388–1389
 project templates (VSTO), 1386–1387
 SharePoint workflow templates (VSTO), 1388

ternary operator (C++/CLI), 1694

Test Client (WCF), 1461–1462

text
 drawing (GDI+), 1123–1125
 editor, VS Studio 2008, 402
 files, reading/writing to, 815–818
 properties (Windows Forms elements), 660
 text boxes, binding columns to, 1075–1076
 TextBox controls (Windows Forms),
 1036–1037

textures, adding to triangles, 1241–1242

themes (ASP.NET)
 applying to pages, 1339–1340
 defining, 1340
 in PCSDemoSite, 1340–1344

themes in Workflow Designer, 1498–1500

this keyword, 110

Thread class
 background threads, 541–542
 basics, 535, 538–540
 controlling threads, 543
 creating threads, 535, 538–540
 passing data to threads, 540–541
 thread priority, 542–543

threading

- asynchronous delegates, 535–538
 - COM and, 753–754
 - COM apartments, 569–570
 - COM components, 764–765
 - deadlock, 548–549
 - event-based asynchronous pattern. See BackgroundWorker class; event-based asynchronous pattern
 - issues, 533, 545
 - messages and, 1573–1574, 1581
 - overview, 533–535
 - race conditions, 545–548
 - reasons for using, 533
 - synchronization. See synchronization
 - thread class. See thread class
 - thread, defined, 534
 - thread pools, 543–545
 - threads, starting, 178
 - timers, 568–569
 - Windows Services and, 730
- throwing exceptions, 380**
- throwing user-defined exceptions, 391–394**
- tiered development (ADO.NET), 889–890**
- time quantum (threads), 542**
- timelines, 1228–1232**
- Timer control (ASP.NET Ajax), 1360**
- timers, 568–569**
- tlbimp utility, 762**
- toolbars, creating (WebBrowser control), 1433–1437**
- toolbox, Visual Studio 2008, 420–421**
- ToolboxBitmap attribute (WF), 1500**
- ToolStrip control (Windows Forms), 1039–1041**
- ToolStripContainer control (Windows Forms), 1042**
- ToolStripManager class (Windows Forms), 1042**
- ToolStripMenuItem class (Windows Forms), 1042**
- ToString() method**
- date formatting and, 644–645
 - dictionary example, 280–281
 - fundamentals/examples, 98–99
 - number formatting and, 643–644
 - purpose of, 97

tracing

- architecture of, 510
- asserts, 516–517
- basics, 509–510
- defined, 509
- filters, 515–516
- trace listeners, 510, 512–515
- trace sources, 510–511, 1466–1467
- trace switches, 512

tracking

- object, 1674–1675
- service (WF), 1518–1520
- track points (WF), 1519

transactions

- ACID properties, 681
- ADO.NET, 683–684, 905
- ambient, 694–701
- committable, 687–690
- CourseManagement sample database, 681–683
- custom resource managers, 703–709
- database and entity classes, 681–683
- database connections and, 854–855
- dependent, 692–693
- Enterprise Services, 684–685
- isolation levels and, 701–703
- overview, 679–680
- phases, 680–681
- promotable, 690–692
- System.Transactions namespace, 685–687
- transactional messages, 1559
- transactional queues, 1577–1578
- transactional resources, 704–709
- TransactionOption values, 1539
- TransactionScope class members, 694
- TransactionScopeOption enumeration, 696
- with Windows Vista/Windows Server 2008, 710–713

transactions, Enterprise Services

- automatic, 1529, 1539
- basics, 1539–1540
- configuring with Component Services Explorer, 1537
- distributed, 1530
- promotable, 1530

transformation elements (WPF), 1162–1163

transforms, coordinate (GDI+), 1137–1138

translations, outsourcing, 666–667

Transmission Control Protocol (TCP). See TCP (Transmission Control Protocol)

TreeView-based custom control (Windows Forms), 1049–1055

triangle models (WPF)

- 3-D objects and, 1242–1245
- basics, 1238–1240
- lights, changing, 1240
- textures, adding, 1241–1242

triggers

- defined, 1001
- event triggers (animations), 1233–1235
- messaging queuing and, 1558
- property in WF, 1182–1184
- SQL Server and, 1001–1003
- TriggerAction classes, 1236
- UpdatePanel triggers, 1371–1372

troubleshooting Windows Services, 745–746

True Type font families, 1126

truncating numbers, 168

trust licenses, 460

Trusted Application Deployment, 460

try blocks

- defined, 379–380
- examples, 382–386
- nested, 387–389
- try . . . catch . . . finally blocks, 852

TryGetValue() method, 282

tunneling events, 1177

two-way binding, 1200, 1204

type class

- properties and methods, 365–367
- types defined in assemblies, 370
- TreeView example, 367–370

type safety

- boxing and unboxing, 152–153
- overview, 148
- type conversions, 149–151

types

- anonymous, 91–92
- basics, 224
- collections, 247–250
- condition types (WF), 1490
- constraint types (generics), 233
- conversion (lists), 260–261
- filtering, 312–314
- generic type names, 226
- integer types, and pointers, 343

key types (dictionaries), 278–280

library, creating, 772–774

managed/unmanaged, 343

metadata, 470–471

modifying exception, 387–389

passed with contracts, 1257

safety and generics, 223, 225

typeof operator, 146

TreeView example, 367–370

user-defined. *See* user-defined types (UDTs)

types, defining in .NET languages

enumerations, 1686

interfaces, 1685–1686

reference types, 1683–1684

type inference, 30–31, 1685

value types, 1684–1685

U

UAC (user account control), 1712–1715

UDP (User Datagram Protocol)

class, 1448–1449

vs. TCP, 1448

UDTs (user-defined types). See user-defined types (UDTs)

UI customization (VSTO), 1401–1405

unboxing, defined, 224

unhandled exceptions, 386–387

Unicode

globalization and, 640–641

text files with, 651

unmanaged resources, freeing, 334–339

destructors, 335–339

IDisposable interface, 336–339

unmanaged types, defined, 343

unsafe code blocks (C#), 1709

unsafe keyword, 340–341

unsafe methods, 340–341

unsafe safety level, 986

updating

ADO.NET Entity Framework, 1674–1678

records, 860–861

update and delete constraints, setting, 876

UpdatePanel control (ASP.NET Ajax), 1360, 1370–1372

UpdateProgress control (ASP.NET Ajax), 1360, 1372

UpdateRowSource values, 886

uploading files with WebClient class, 1426

URIs (Uniform Resource Identifiers)

classes, 1440–1441

vs. URLs, 1424

user account control (UAC), 1712–1715

user controls (ASP.NET)

basics, 1312–1313

in PCSDemoSite, 1318

simple, 1313–1318

user controls (Windows Forms), 1055–1059

User Datagram Protocol (UDP). See UDP (User Datagram Protocol)

user-defined casts

between base and derived classes, 169–170

boxing/unboxing casts, 170–171

converting between classes, 168–169

fundamentals, 164–165

multiple casting, 171–175

SimpleCurrency example, 165–168

user-defined exceptions

catching, 390–391

defining classes, 394–396

SolicitColdCall example, 389–390

throwing, 391–394

user-defined functions (SQL Server), 1000–1001

user-defined types (UDTs), 988–997

Ajax Library, 1377–1378

basics, 988

creating, 988–993

user-defined aggregates, 996–997

using, 993–996

users

account management, 1619–1620

input, responding to (GDI+), 1138–1141

interaction events (Control class), 1024–1025

interface (Active Directory), 1611–1612

interface customization (VSTO), 1401–1405

User Interface editor (VS 2008), 462–463

user objects, searching for, 1614–1616

user policy level (security), 621–623

user principal name (UPN) (Active Directory), 1601

usernames (Active Directory), 1601

Users and Computers snap-in (Active Directory), 1594–1595

using block statement, 852–854

using directive, 53–54

using keyword, 54, 1682

using statement in .Net languages, 1702

V

validation

activity validation (WF), 1497–1498

validation Web server controls (ASP.NET), 1288–1289

with XmlReader, 928–930

validation, binding

custom validation rules, 1223–1224

data error information, 1221–1222

handling exceptions, 1219–1221

Validation class, 1221

value types

basics, 35–39, 224

comparing for equality, 154

declaring with .NET languages, 1684–1685

memory management and, 330–331

and reference types, 8

structs as, 93–94

values. See also default values (generic classes)

BitVector32, 291

Name/Text (Windows Forms elements), 660

row state, 1066–1067

VALUE index, 1011

Value property (WPF binding), 1201

XML, reading, 1005–1008

var keyword, 30, 91, 1685, xlv

variables. See also constants

constant variables, 34

declaring, 29

implicitly typed (.NET 3.5), xlv–xlvi

initialization of, 29–30

monitoring with Autos window, 432–433

type inference, 30–31

variable scope, 31–34

VARIANT data types, 753

VBA interoperability (VSTO), 1415–1418

Vector struct example (operator overloading), 157–163

VectorClass assembly example, 362, 364–365

versioning

add-ins, 1252, 1259

application configuration files, 500–503

obtaining versions programmatically, 500

- overview, 499
- publisher policy files, 504–506
- runtime version, 506–507
- version dependencies, 470
- version numbers, 499–500
- WCF, 1468–1469
- viewing assemblies, 473**
- viewstate fields (ASP.NET), 1275**
- virtual addressing, defined, 330**
- virtual Equals() method, 153**
- virtual keyword, 104, 106**
- virtual memory, defined, 330**
- virtual methods, 104**
- VirtualizingStackPanel (WPF), 1172**
- visibility modifiers, 112–113**
- Vista, Windows**
 - Advanced Query Syntax (AQS), 1728–1729
 - command link controls, 1716
 - directory structure, 1715–1716
 - file dialogs, 1721–1722
 - OLE DB provider, 1724–1727
 - searching in, 1722–1729
 - task dialog, 1718–1721
 - transactions with, 710–713
 - user account control (UAC), 1712–1715
 - Vista Bridge, 1711, 1717
- Visual Basic 2008**
 - COM components, building with VB 9, 756
 - error handling in VB 6, 377
 - LINQ to SQL and, 967–968
 - mapping syntax from C#, 1681–1709
 - .NET Framework and, 5–6
 - Visual Studio 2008 and, 416
- Visual C++**
 - .NET Framework and, 6
 - Visual Studio 2008 and, 416
- Visual Studio 2008**
 - BasicForm Windows project, 413–415
 - breakpoints, 431–432
 - building/compiling/making, defined, 427
 - C# code/compilation options, 408–410
 - Class View window, 424
 - code optimization, 427–428
 - conditional compilation, 428
 - configurations, editing, 429–430
 - configurations, selecting, 429
 - Console Application, 410–412
 - Conversion Wizard, 404–405
 - Custom Actions editor, 463–465
 - debug and release builds, 427
 - debugger symbols, 428
 - debugging, 430–434
 - deployment projects. *See* deployment Design View window, 419–422
 - exceptions, 433–434
 - extra source code debugging commands, 428–429
 - features and overview, 401–406
 - File Types editor, 461–462
 - folding editor, 416–418
 - installer project types. *See* installer project types
 - languages and, 663
 - Launch Conditions editor, 465–466
 - LINQ to SQL and, 897–902
 - multi-targeting .NET Framework, xlvii, 436–438
 - .NET environment and, xlv
 - new project, creating, 406–407
 - O/R Designer. *See* O/R Designer
 - Object Browser window, 425–426
 - pin buttons, 426
 - project type, selecting, 407–410
 - Properties window, 422–424
 - refactoring applications, 434–436
 - Registry editor, 461
 - Server Explorer window, 426
 - solutions and projects, 412–415
 - Start Page, 407
 - startup project, setting, 415
 - toolbox, 420–421
 - User Interface editor, 462–463
 - Visual Studio Add Service Reference, 1482
 - VS 6 projects, reading in, 416
 - watches, 432–433
 - WF applications, building in, 439–441
 - Windows application code, 415
 - WPF applications, building in, 438–439
 - XML schemas and, 1012
- Visual Studio 2008 for localization**
 - automatic fallback for resources, 666
 - basics, 659–663
 - culture, changing programmatically, 663–665
 - custom resource messages, 665–666
 - outsourcing translations, 666–667

Visual Studio Tools for Office (VSTO), 1385–1419

- application example, 1405–1415
- building solutions. See solutions, VSTO
- event handling, 1397–1398
- Globals class, 1397
- host items/host controls, 1392–1394
- namespaces, 1391–1392
- Office object model, 1391
- overview, 1386–1390
- project features, 1389–1390
- project structure, 1394–1396
- project types, 1386–1389
- VBA interoperability, 1415–1418

visual styles (Windows XP), 1047

VisualBrush (WPF), 1166–1167, 1242

void pointers, 344

vtable (virtual table), 751

W

W3C standards supported in .NET, 922

wait handles

- asynchronous delegates and, 536–537
- mutex and, 559
- synchronization and, 557–558

#warning preprocessor directives, 65

WAS (Windows Activation Services) hosting, 1481–1482

watches (debugging), 432–433

WCF (Windows Communication Foundation), 1455–1486

- binding, 1477–1480
- clients, 1482–1483
- contracts, 1467–1471
- duplex communication, 1484–1486
- error handling, 1476–1477
- hosting, 1480–1482
- integration with WF, 1521–1524
- JSON, 1458
- message queuing. See message queuing with WCF
- overview, 22, 1456–1458
- service implementation, 1471–1477
- settings, 496
- SOAP, 1457
- WSDL, 1457–1458

WCF, integrating with Enterprise Services

- client application, 1553–1554
- WCF service façade, 1549–1553

WCF, service example

- custom Service Host, 1463–1464
- diagnostics, 1466–1467
- getting started, 1458–1459
- implementation, 1460–1461
- service contract, 1460
- Service Host and Test Client, 1461–1462
- WCF client, 1464–1465

Web applications

- configuring with Web.config files, 1308–1309
- Xcopy and, 447

Web browsing from applications, 1430–1431

Web Forms (ASP.NET), 1275–1294

- code model, 1278–1279
- controls, defined, 420
- defined, 1273
- overview, 20, 1275–1278
- server controls. See server controls (ASP.NET)

Web method calls (ASP.NET AJAX), 1381–1383

Web pages

- accessing master pages from, 1325–1326
- applying themes to, 1339–1340
- ASP.NET support for, xliiv

Web Parts (ASP.NET), 1344–1353

- basics, 1344
- controls, 1344–1346
- example, 1346–1353

Web server controls (ASP.NET)

- basics, 20, 1279
- data, 1287–1288
- data binding and, 1302–1303
- example, 1289–1294
- standard controls, 1283–1286
- validation controls, 1288–1289

Web services

- ASP.NET application services, 1383
- Web Services Description Language (WSDL), 1457–1458

Web Setup installer Project, 448

Web sites

- configuration example (ASP.NET Ajax), 1365–1368

- publishing, 444, 447–448
- simple example (ASP.NET Ajax), 1362–1364
- Web sites, for downloading**
 - Ajax Library, 1358
 - ASP.NET, 1358
 - ASP.NET Ajax Control Toolkit, 1358
 - DSML, 1620
 - Nikhil's Web Development Helper, 1381
 - PNRP, 1629
 - Reflector software, 853, 1500
 - source code for this book, li
 - Unicode example, 821
- Web sites, for further information**
 - ASP.NET, 1358
 - culture name definitions, 641
 - database connection strings, 850
 - P2P forums, lii
 - Workflow Designer, 1526
- WebBrowser control**
 - HTML pages and, 1429–1430
 - Internet Explorer features and, 1432–1433
 - for printing, 1437
 - requested-page code, displaying, 1438
 - Web browsing from applications, 1430–1431
- WebClient class, 1424–1426**
- Web.config files, 1308–1309**
- WebRequest/WebResponse classes**
 - basics, 1426–1429
 - hierarchy, 1438–1440
- WelcomeMessage() method, 660–661, 665**
- WF (Windows Workflow Foundation), 1487–1526**
 - activities. See activities (WF)
 - applications, building in VS, 439–441
 - basics, 1487
 - Designer, 1487, 1526
 - Hello World example, 1488–1489
 - hosting workflows, 1524–1525
 - integration with WCF, 1521–1524
 - workflow runtime, 1514–1515
 - workflow services. See services, workflow (WF)
 - workflows. See workflows (WF)
- WhatsNewAttributes example, 362–365, 371–375**
- Where() extension method, 305–308, 312**
- while/do . . . while Statements, 1696–1697**
- while loops, 48**
- WinCV utility, 18**
- windows**
 - drawing scrollable, 1107–1113
 - pinning, 426
- Windows, Microsoft**
 - Activation Services (WAS) hosting, 1481–1482
 - application code, 415
 - Authentication (ASP.NET), 1331
 - Communication Foundation (WCF). See WCF (Windows Communication Foundation)
 - Controls, defined, 21
 - Forms. See Windows Forms
 - history of, xliii
 - Installer. See Windows Installer
 - Presentation Foundation (WPF). See WPF (Windows Presentation Foundation)
 - Resource Localization Editor, 667
 - Server 2008. See Windows Server 2008
 - Services. See Windows Services
 - Task Manager, 534
 - Vista. See Vista, Windows
 - WindowsIdentity class, 583
- Windows Forms**
 - ActiveX Control Importer, 768
 - ActiveX controls in, 768–770
 - application, creating, 768–770, 1018–1023
 - Button class, 1026–1027
 - CheckBox control, 1028
 - CheckedListBox control, 1029–1031
 - class hierarchy, 1023
 - ComboBox control, 1029–1031
 - ContextMenuStrip class, 1042
 - control class. See control class (Windows Forms)
 - controls, defined, 420
 - controls in Internet Explorer, 782–783
 - creating, 21
 - custom controls. See custom controls (Windows Forms)
 - DateTimePicker control, 1031
 - Designer, 661, 667, 769
 - ErrorProvider component, 1031–1032
 - FlowLayoutPanel container, 1037–1038
 - form appearance, 1046–1047
 - Form class, 1043

Windows Forms (continued)

- form instantiation and destruction, 1043–1046
- HelpProvider component, 1032
- ImageList component, 1033
- labels control, 1033
- ListBox control, 1029–1031
- ListView control, 1033–1035
- localization and. See Visual Studio 2008 for localization
- MDI-type applications, 1047–1048
- MenuStrip control, 1041–1042
- overview, 1017
- Panel control, 1037
- PictureBox control, 1035
- ProgressBar control, 1035–1036
- radio button control, 1028
- SplitContainer control, 1038
- TabControl/TabPage controls, 1038–1039
- TableLayoutPanel container, 1037–1038
- TextBox controls, 1036–1037
- ToolStrip control, 1039–1041
- ToolStripContainer control, 1042
- ToolStripManager class, 1042
- ToolStripMenuItem class, 1042
- WPF and, 1245–1248

Windows Installer

- vs. ClickOnce, 458
- fundamentals, 449
- Windows Installer Search, 465

Windows Server 2008

- transactions with, 710–713
- user account control and, 1712–1715

Windows Services

- application, creating, 725–729
- class library using sockets, 719–722
- controlling with Server Explorer, 737
- defined, 21–22
- event logging and, 746
- fundamentals, 715–716
- handler methods, 728–729
- installer classes, 730–734
- installutil.exe utility, 734
- interactive services, 745–746
- main function, 727–728
- MMC Services snap-in, 735–736
- net.exe command-line utility, 734–736
- power events and, 746–747

- sc.exe command-line utility, 734–737
- service programs, 716–718
- ServiceBase class, 726–727
- ServiceController class application, 737–745
- System.ServiceProcess namespace, 718–719
- TcpClient example, 722–725
- threading and, 730
- troubleshooting, 745–746

Windows Workflow Foundation (WF). See WF (Windows Workflow Foundation)

Word documents

- vs. Excel document hosting, 1405
- host items/controls for (VSTO), 1392–1393
- storing as XML, 1003
- template example (VSTO), 1394–1395
- WordDocEditTimer, 1405–1415

workflows (WF)

- binding parameters to activities, 1513–1514
- defined, 1489, 1508
- passing parameters to, 1511–1512
- queue, 1492
- returning results from, 1512–1513
- runtime, 1514–1515
- sequential, 1509
- services. See services, workflow (WF)
- state machine, 1509–1511
- support for, 1488
- template, 1487
- Windows Workflow Foundation (WF). See WF (Windows Workflow Foundation)
- WorkflowInstance, 1487
- WorkflowRuntime code, 1487
- XML-based, 1525

world coordinates (GDI+), 1113–1114

WPF (Windows Presentation Foundation), 1199–1249

- 3-D features in. See 3-D features in WPF; triangle models (WPF)
- animations. See animations
- applications, building in VS, 438–439
- attached property, 1153–1154
- basics, 21
- brushes, 1163–1167
- class hierarchy, 1155–1156
- command bindings, 1224–1228
- content controls, 1168–1169
- data binding. See data binding and WPF
- dependency property, 1152–1153

designer/developer cooperation, 1154–1155
 event handling, 1176–1177
 headered content controls, 1170–1171
 headered items controls, 1171–1172
 items controls, 1171
 layout containers, 1172–1176
 ListBox, styling, 1192–1197
 localization with, 669–673
 markup extensions, 1154
 namespaces, 1156–1159
 resources, 1178–1181
 shapes, 1159–1162
 simple controls, 1167–1168
 styles, defining, 1177–1178
 Syndication Reader and, 1645
 templates, 1184–1192
 transformation elements, 1162–1163
 triggers, 1182–1184
 Windows Forms and, 1245–1248
 WPF Browser Application, 1249
 WrapPanel container, 1173
 XAML. *See* XAML (XML for Applications Markup Language)

write-only properties, creating, 84

WriteAllBytes() method, 806–807

WriteAllLines() method, 806–807

WriteAllText() method, 807

WriteEntry() method, 520, 522, 525

WriteEvent() method, 520, 522, 525

writing

to files, 805, 807–808
 to files using FileStream class, 810–812
 to isolated storage, 839–844
 to registry. *See* registry
 to text files, 815, 818–819

WSDL (Web Services Description Language), 1457–1458

X

XAML (XML for Applications Markup Language)

code for WPF application example, 670
 data binding with, 1200–1203
 defining instances with, 1206–1207
 fundamentals, 1150–1152
 localization with, 671–673
 properties as attributes, 1152

properties as elements, 1152

XAttribute object, 974

XBAP files, defined, 1249

XComment object, 973

Xcopy deployment, 444, 447

XDocument object, 969

XElement object, 969–970

XML (Extensible Markup Language)

Ajax and, 1357
 automatic documentation in, 61–62
Beginning XML (Wiley), 921
 converting ADO.NET data to, 948–954
 converting to ADO.NET, 954–956
 data binding to, 1217–1219
 DML (XML Data IModification Language), 1010
 DOM implementation in .NET, 931–932
 LINQ to XML. *See* LINQ to XML
 namespaces and prefixes, 968
 .NET Framework and, 921
 output, writing, 887–889
 outputting from database query, 982–983
 populating DataSets from, 884
 serializing objects in, 956–965
 site maps, 1328–1329
 streamed XML, reading/writing, 924
 System.Xml classes, 923–924
 System.Xml namespace, 922–923
 System.Xml.Xsl namespace, 942
 transforming, 943–947
 W3C standards supported in .NET, 922
 workflows and, 1525
 XML Web services, overview, 20
 XmlCharacterData classes, 932
 XmlDocument class, 933–936, 1006–1007
 XMLHttpRequest (Ajax), 1357
 XmlLinkedNode classes, 932
 XmlNode classes, 931–932
 XmlReader class, 924–928, 1005–1007
 XmlReader, validating with, 928–930
 XmlWriter Class, 930–931

XML data type

overview, 1003
 query of data, 1008–1010
 strongly typed XML, 1011–1013
 tables with XML data, 1003–1005
 XML DML, 1010
 XML indexes, 1010–1011
 XML values, reading, 1005–1008

XML documents

XML documents

- querying dynamic, 976–978
- querying static, 975–976
- reading from, 978–979
- writing to, 979–981

XML schemas

- strongly typed XML and, 1011–1013
- Word/Excel support, 1003
- XML schema definitions (XSD), 877–883

XNamespace object, 971–972

XPath

- reading XML values and, 1006
- XML indexes and, 1011
- XML queries and, 1009

XPathNavigators

- System.Xml.XPath namespace, 937–942
- XPathDocument class, 937
- XPathNavigator class, 937–939
- XPathNodeIterator class, 939

XQuery

- basics, 1008–1010

XML data type and, 1003

XSD (XML schema definitions), 877–883

XSL (Extensible Stylesheet Language), 942–947

XSLT (Extensible Stylesheet Language Transformation)

- debugging, 947–948
 - XsltArgumentList, 944–946
- x:Uid attribute, 671–673**

Y

yield statements

- defined in C#, 1709
- enumerations and, 134–139

Z

Zone membership condition, 614