

# Index

---

**Note to the Reader:** Page numbers in **bold** indicate the principle discussion of a topic or the definition of a term. Page numbers in *italic* indicate illustrations.

## A

---

- Activator class, 168–169
- Add New Project dialog box, 404, 404
- ADO
  - versus ADO.NET, 337
  - Data Control. *See* data binding
  - database connections, 330–331
  - RecordSets, 336–337, 342, 343, 346
- ADO.NET database connections, **326–336**
  - connection pooling
    - auto garbage collection and, 331–332
    - customizing, 333
    - defined, **332–333**
    - Max Pool Size property, 334
    - monitoring, 334–336, 335
    - open connections and, 331
    - overview of, 330–331
    - warning, 335
  - overview of, 326, 336
  - using Server Explorer GUI
    - accessing, 326–327
    - adding connection monitors, 336
    - creating connections, 327, 327, 330
    - drilling into connections, 328, 329
  - reusing connection strings, 330
    - testing connections, 328, 330
    - viewing code behind, 329–330
  - versus in VB6 and classic ADO, 330–331
- ADO.NET DataSets, **336–368**
  - versus ADO RecordSets, 336–337, 342, 343, 346
  - ADO.NET, overview, 339
  - calculated DataColumnns
    - adding columns, 356, 356
    - aggregating data, 357–358, 357
    - bug, 359
    - defined, **353**
    - expression syntax, 358–359
    - loading data, 354–356, 356
    - normalization and, 353
    - performance, 359
    - warning, 358
    - when to use, 354, 359
  - distributed applications and, 412
  - Merge method
    - creating test tables, 361–363
    - defined, **360**
    - using with GetChanges, 366–368
    - merging schemas with data, 364–366, 365
    - merging two tables, 362–364, 365
    - minimizing data transfers, 366–368
    - MissingSchemaAction values, 366
    - overloads for, 360–361
    - PrimaryKey property and, 362–363
    - saving merged data, 366
    - uses for, 368
  - overview of, 336–337, 346, 403
  - ReadXml method, 387
  - versus relational databases, 354, 360
  - synchronizing with DiffGrams
    - code behind, 409–412
    - creating client applications, 406–407, 406
    - creating Web services, 404–406, 404
    - defined, **403**
    - disconnected editing, 408, 409
    - GetChanges method, 411
    - minimizing data transfers, 403, 411
    - overview of, 380
    - saving changes, 408, 410, 412
    - SQLXML library support, 370
    - testing, 408, 408
    - Web references and, 406–407, 406
    - WriteXml method, 411
  - typed DataSets
    - benefits of, 352–353
    - creating via drag and drop, 347–348, 349
    - creating with Schema Designer, 347, 348–351, 350
    - defined, **347**
    - versus untyped DataSets, 353
  - untyped DataSets
    - adding data, 344

- in ADO.NET architecture, 338, 338
- CommandBuilder object and, 345–346
- creating table relations, 341–342, 343
- DataAdapter object and, 345, 346
- defined, 337
- deleting data, 344–345
- editing data, 343–345
- late-bound syntax, 347
- loading data tables, 339–341, 341
- navigating, 342–343, 343
- saving changes, 345–346
- versus typed DataSets, 353
- warning, 346
- WriteXml method, 387, 411
- XmlDataDocuments and
  - creating DataSets from, 134–135
  - creating from DataSets, 132–134, 386
  - creating schema-only DataSets and, 386–387
  - defined, **132, 380, 384**
  - displaying as DataSets on DataGrids, 385, 386
  - overview of, 380
  - versus ReadXml method, 387
  - relational data and, 132, 387
  - using synchronized objects, 387
  - synchronizing, 384–387, 386
  - when to use, 132
  - versus WriteXml method, 387
  - XmlTextReader/Writer classes and, 385
  - in XPath data queries, 387
  - in XSLT data formatting, 387
- XmlDocuments and
  - Document Object Model, 380–382, 382
  - loading XML into, 382–384, 383–384
  - overview of, 380
  - XmlNode class and, 382–383, 383–384
  - XmlTextReader/Writer classes and, 384
- API INI functions, **28–37, 38**
- API ShellExecute function, **187–189, 196**
- application configuration files, **151–152, 153**, *See also* storing
- Apra, Victor Garcia, 318
- ArrayLists
  - adding events to, 174–177
  - versus ValueCollections, 180, 186
  - wrapping in CollectionBase class, 172–174
  - wrapping in custom classes, 170–172
- arrays. *See* conversions; sort
- articles online, 42, 58, 113
- ASCII values in conversions, 157, 160, 161
- ASP.NET applications, **254–324, 412–421**
  - custom configuration handlers
    - configuration sections and, 255–256, 255
    - for custom tags, 258–261
    - in machine.config file, 256, 257, 264
    - NewLine values shortcut, 264
    - Response.Output.Writeline method, 264
    - System handlers for custom tags, 256–258
    - testing CustomItemHandlers, 262–264, 263
    - warnings, 259, 262
      - in web.config file, 254–259, 255, 262, 263, 264
  - data binding, 413
  - defining modal message box controls
    - adding to Toolbox, 321
    - client-side message boxes, 321–322, 321, 323
    - in client-side script, 307–309
    - at design time, 309–313, 320–322, 321
    - goals to be met by, 309
    - how they work, 314–316
    - initializing, 316–320
    - in JavaScript, 308–309, 314
    - message box features, 306–307, 307
    - overview of, 306, 309–310, 324
    - properties, 310–314
    - Render method workaround, 318
    - rendering design-time HTML, 312, 315, 320
    - at runtime, 310, 314, 322, 323
    - server-side message boxes, 322, 323
    - using showModalDialog method, 309, 313, 314, 316, 317
    - in VBScript, 307–308, 309, 313, 314, 317
    - versus Windows Forms message boxes, 306–307, 307
  - internationalizing
    - caching article content, 285–286
    - creating images dynamically, 287–290, 288, 291
    - improving culture matching, 292–293
    - language/culture issues, 266–267
    - letting users choose languages, 274–276, 276
    - localizing database content, 291
    - localizing dates, 290
    - localizing numbers/currencies, 290–291
    - localizing static images, 286–287, 287–288, 291–292
    - localizing text resources, 277–280, 277–278, 280, 292
    - managing content, 293
    - managing resource changes, 292
    - mapping content to classes, 267–274, 270, 273–274
    - overview of, 265
    - ResourceManager objects and, 292
    - specifying cultures, 281–284, 284
  - paging data in DataGrid controls
    - adding paging controls, 419–420
    - using automatic paging, 414–416, 415–417, 420, 421
    - using custom paging, 416–420, 421, 421
    - customizing paging controls, 415–416
    - data-binding code, 420
    - versus not paging, 413, 414, 420
    - overview of, 412–413, 420–421
    - stored procedures in, 417–418
    - warning, 420
  - Web Forms custom FocusManager control

- adding properties, 296–297
- adding to Toolbox, 299–300, 300
- creating, 294–296
- overriding Render method, 297–299
- overview of, 294
- testing, 300–304, 305

## B

---

- base Object class, To String method, 159, 167
- base type array sorting, 100–101, 101
- base types, conversions between SqlDbTypeTypes and, 162–163
- binary file I/O. *See* file I/O
- BindingContext objects, *See also* data binding
  - in data-binding example, 398–400, 398
  - defined, 397
  - PropertyManager class and, 402
- browser applications. *See* ASP.NET
- buffer size, setting for StreamReader, 85–86
- byte arrays
  - avoiding conversions of, 70–71, 72
  - converting strings to, 160, 161
  - converting to strings, 161–162
- byte-to-char/char-to-byte conversions, 157, 157–159

## C

---

- calculated DataColumn, *See also* ADO.NET DataSets
  - adding columns, 356, 356
  - aggregating data, 357–358, 357
  - bug in, 359
  - defined, 353
  - expression syntax, 358–359
  - loading data, 354–356, 356
  - normalization rules and, 353
  - performance issues, 359
  - warning, 358
  - when to use, 354, 359
- characters, *See also* regular expressions
  - character sets, 157
  - converting strings to Char arrays, 159
  - converting to/from bytes, 157–159, 157
- classes, adding sort capabilities to
  - built-in QuickSort method, 100, 109
  - considering efficiency, 109
  - using IComparable interface, 100–104
  - using IComparer interface, 104–106
  - sorting
    - base type arrays, 100–101, 101
    - custom class arrays, 102–105
    - custom collections, 106–107
    - multidimensional arrays, 107–109
    - partial arrays, 105
    - typed arrays, 105–106
    - warning, 101
- cloning objects, 79, 80
- CollectionBase class, 172–174
- collections, *See also* objects
  - custom, sorting, 106–107
  - deserializing, 77
  - persisting, 76–77
  - serializing, 76–77
- collections, building custom type-safe, 170–185
  - accessible by index or key, 172
  - case-sensitive string collections, 172
  - using containment, 170–172
  - inheriting from
    - CollectionBase, 172–177
    - CollectionsUtil, 172
    - DictionaryBase, 172, 177–179
    - NameObjectCollectionBase, 172
    - ReadOnlyCollectionBase, 172
  - Object type and, 170, 179
  - overview of, 170
  - that sort by keys, 172
  - value type collections, 179–186
- color objects, creating from strings, 167–168
- columns. *See* calculated DataColumn
- ComboBoxes, drawing custom, *See also* data-bound
  - displaying font names/styles, 25–27, 27
  - listing files/folders/icons, 17–23, 17
  - overview of, 16
  - with variable width/height items, 23–25, 23
- command lines, parsing/validating, 212–234, *See also* console applications
  - command lines
    - defined, 212
    - flag parameters, 212–213
    - optional parameters, 212, 213
    - splitting, 214
  - extending parsers, 220–234
  - generic guidelines for, 214–215
  - how parsers work, 215–217, 217
  - overview of, 212
  - setting up parsers, 217–219, 219–220
  - warning, 216
- CommandBuilder object, 345–346
- comments, in upgrading INI files to XML, 47
- compatibility syntax. *See* file I/O
- configuration files, 151–152, 153, *See also* storing
- configuration handlers, creating in ASP.NET, 254–264
  - configuration sections and, 255–256, 255
  - custom handlers for custom tags, 258–261
  - in machine.config file, 256, 257, 264
  - NewLine values shortcut, 264

Response.Output.WriteLine method and, 264  
 System handlers for custom tags, 256–258  
 testing CustomItemHandlers, 262–264, 263  
 warnings, 259, 262  
 in web.config file, 254–259, 255, 262, 263, 264

console applications, *See also* command lines  
 InstallUtil.exe, 244–245, 245  
 using touch utilities from, 205–211, 205

containment, **170–172**

conversions, **154–169**  
 using Activator class, 168–169  
 ASCII values in, 157, 160, 161  
 byte arrays to strings, 161–162  
 byte-to-char/char-to-byte, 157–159, 157  
 using Convert class, 154–156, 156, 169  
 using IConvertible interface, 165  
 Integer arrays to strings, 162  
 numeric values in, 157–158  
 between reference types, 163–167  
 rules for, 169  
 between SqlDbTypeypes and base types, 162–163  
 strings to byte arrays, 160, 161  
 strings to Char arrays, 159  
 strings to objects, 167–169  
 using TypeConverter class, 165–166, 167–168, 169  
 Unicode values in, 157, 158, 160, 161  
 value types to strings, 159

Cornes, Ollie, 265

CurrencyManager objects, *See also* data binding  
 class members, 401–402  
 in data-binding example, 398–400, 398  
 defined, 397  
 events, 401

custom class array sorting, **102–105**  
 custom file formats, storing data in, **141–143**, 153

---

## D

data binding  
 in ASP.NET, 420  
 BindingContext objects  
 defined, 397  
 example, 398–400, 398  
 PropertyManager class and, 402  
 CurrencyManager objects  
 class members, 401–402  
 defined, 397  
 events, 401  
 example, 398–400, 398  
 .NET architecture of, 397–398, 398  
 the .NET way, 396–397, 402–403

Data Control, ADO. *See* data binding

data transfers, minimizing, 366–368, 403, 411

data-bound data-input forms  
 adding ComboBox controls, 391–392, 392  
 adding ListBox controls, 392–393, 393  
 binding to data coded into, 394–396  
 binding to DataSets, 391–392, 392, 393  
 using bound ListBoxes/ComboBoxes, 388, 396  
 building, 389–390, 390  
 overview of, 388  
 revising code behind, 393–394  
 using Server Explorer, 389–390, 390  
 stored procedures, 389, 390  
 testing, 391, 391, 394

DataAdapter object, 345, 346, 410

databases, *See also* ADO.NET  
 normalization, 353  
 relational, versus DataSets, 354, 360  
 relational, XmlDataDocuments and, 132, 387  
 storing data in, 153

DataGrids, paging ASP.NET data in, **412–421**  
 adding paging controls, 419–420  
 using automatic paging, 414–416, 415–417, 420, 421  
 using custom paging, 416–420, 421, 421  
 customizing paging controls, 415–416  
 data-binding code, 420  
 versus not paging, 413, 414, 420  
 overview of, 412–413, 420–421  
 stored procedures in, 417–418  
 warning, 420

DataLink Properties dialog box, 327, 327

DataSets. *See* ADO.NET DataSets

dates, finding in strings. *See* regular expressions

dates/times, altering. *See* touch utilities

default namespace prefixes, creating, 125

deleting DataSet data, 344–345

deserializing collections, 77

deserializing objects, 72, 76, 77

DictionaryBase collections, 172, **177–179**

DiffGrams, synchronizing with DataSets, *See also* ADO.NET  
 DataSets  
 code behind, 409–412  
 creating client applications, 406–407, 406  
 creating Web services, 404–406, 404  
 defined, 403  
 disconnected editing, 408, 409  
 GetChanges method, 411  
 minimizing data transfers, 403, 411  
 overview of, 380  
 saving changes, 408, 410, 412  
 SQLXML library support, 370  
 testing, 408, 408  
 Web references and, 406–407, 406  
 WriteXml method, 411

DisplayRules, *See also* TreeView  
 creating, 54–55  
 customizing TreeViewXml, 60–62, 60–61

properties, 54  
 types of, 53  
 XSLT option, 54–55, 56, 58, 60–62, 62  
 distributed applications. *See* DiffGrams  
 DOM (Document Object Model), **380–382**, 382  
 downloading Internet files, 77–78, 78

---

## E

editing  
 DataSets, 343–345  
 DiffGrams and, 408, 409  
 resource files, 277, 277  
 EmbeddedResourceReader, 112–113  
 embedding resource files in .NET code, 58, **110–113**  
 enumerations, **23**, 68  
 evidence, 146  
 expressions. *See* regular expressions  
 extending command line parsers, **220–234**  
 extending TreeViewXml control, **64**  
 extensibility in upgrading INI to XML, **46–47**  
 external programs, launching/monitoring, **186–199**  
 overview of, 186–187  
 using Process class  
 example, 189–190  
 launching invisible processes, 194  
 launching processes, 192–193  
 monitoring launched processes, 192  
 overview of, 190–191  
 redirecting process IO, 196–199  
 retrieving invisible process output, 194–195  
 setting parameters at design time, 191, 191  
 waiting for launched process exit, 192–193, 193  
 in VB6, 187  
 using VB.NET Shell function, 187, 199  
 using Win32 ShellExecute function, 187–189, 196

---

## F

file dates/times, altering. *See* touch utilities  
 file formats, custom, storing data in, **141–143**, 153  
 file I/O in VB.NET, **66–86**  
 avoiding compatibility syntax  
 beware of convenience, 85–86  
 FileStream vs. StreamReader/Writer tests, 85  
 overview of, 80–81  
 VB6 versus VB.NET compatibility test, 81–82  
 VB6 versus VB.NET FileStream test, 82–83  
 VB.NET compatibility vs. FileStream test, 84, 84  
 VB.NET compatibility vs. StreamReader/  
 Writer test, 84, 84  
 why compatibility code is slow, 84–85

compatible random access support, 67–68  
 fileMode enumeration, 68  
 using streams and formatters  
 avoiding byte array conversions, 70–71, 72  
 binary file uses, 74  
 BinaryFormatter class, 72, 75–77  
 BinaryReader/Writer classes, 67, 70–71, 72  
 cloning objects, 79, 80  
 converting text files to XML, 73, 73, 75  
 deserializing collections, 77  
 deserializing objects, 72, 76, 77  
 downloading Internet files, 77–78, 78  
 using FileStream class, 67–69, 70  
 using MemoryStream class, 79, 80  
 for more flexibility, 71–72, 72  
 opening files, 68–69  
 optimizing StreamReader speed, 85–86  
 overview of, 66  
 persisting collections, 76–77  
 persisting objects, 73, 73, 75  
 reading/writing text files, 67, 70, 71, 74  
 serializing collections, 76–77  
 serializing objects, 71–73, 72–73, 75  
 setting StreamReader buffer size, 85–86  
 SoapFormatter class, 73, 73, 75  
 StreamReader/Writer classes, 67, 70, 71, 74  
 types of streams, 66–67  
 XmlFormatter object, 73, 73, 75  
 FileSystemWatcher component, 235, **236–240**  
 finding data in strings. *See* regular expressions  
 flag parameters for command lines, **212–213**  
 font names, displaying in font styles, **25–27**, 27  
 formatters. *See* file I/O

---

## G

Generate DataSet dialog box, 328, 330, 391, 392  
 GetChanges method, 366–368, 370, 411  
 Glenwright, Anthony, 58, 113

---

## H

handlers. *See* configuration handlers  
 HTML format, 131–132, 312, 315, 320

---

## I

I/O file. *See* file I/O  
 IComparable interface, **100–104**, *See also* sort  
 IComparer interface, **104–106**

icons, adding to file/folder lists, 17, 17–23

IConvertible interface, 165

INI files

applications, 38

defined, 28

items, 37–38

section/key/values, 37–38

storing data in, 137–139, 138, 152

as ubiquitous, 37–38

upgrading to XML, 28–47

API INI functions, 28–37

comments and, 47

extensibility, 46–47

getting original files back, 47

moving beyond COM Interop, 37

not using .NET configuration files, 38–39

overview of, 28

retrieving values, 42–43

updating/adding values, 44–45, 45

XML case sensitivity and, 43–44

XML INI file structure, 39–42

InstallUtil.exe console application, 244–245, 245

Integer arrays, converting to strings, 162

Integer value type collections, 179–186

Internet files, downloading, 77–78, 78

Isolated Storage, 146–148, 153

ItemData. *See* ListBoxes

## J

JavaScript in ASP.NET message boxes, 308–309, 314

## L

late-bound syntax, 347

launching external programs. *See* external programs

ListBoxes, 2–27, *See also* data-bound

drawing custom

displaying font names/styles, 25–27, 27

listing files/folders/icons, 17–23, 17

overview of, 16

with variable width/height items, 23–25, 23

ItemData gone solution

assigning display methods to instances, 14–15, 15

class creators have control, 6–11, 8, 10–11

delegating control to consumers, 11–14

getting data back, 5–6

mimicking classic VB ListBoxes, 3–11, 8, 10–11

overview of, 2–3

populating ListBoxes, 3–5

## M

machine.config file, 256, 257, 264

Max Pool Size property, 334

MemoryStream class, 79, 80

Merge method of DataSets, *See also* ADO.NET DataSets

creating test tables, 361–363

defined, 360

using with GetChanges, 366–368

merging schemas with data, 364–366, 365

merging two tables, 362–364, 365

minimizing data transfers, 366–368

MissingSchemaAction values, 366

overloads for, 360–361

PrimaryKey property and, 362–363

saving merged data, 366

uses for, 368

message boxes, defining in ASP.NET, 306–324

adding to Toolbox, 321

client-side message boxes, 321–322, 321, 323

in client-side script, 307–309

at design time, 309–313, 320–322, 321

features, 306–307, 307

goals to be met, 309

how they work, 314–316

initializing, 316–320

in JavaScript, 308–309, 314

overview of, 306, 309–310, 324

properties, 310–314

Render method workaround, 318

rendering design-time HTML, 312, 315, 320

at runtime, 310, 314, 322, 323

server-side message boxes, 322, 323

using showModalDialog method, 309, 314, 316, 317

in VBScript, 307–308, 309, 313, 314, 317

versus Windows Forms message boxes, 306–307, 307

minimizing data transfers, 366–368, 403, 411

MissingSchemaAction values, 366

modal dialogs. *See* message boxes

monitoring database connection pooling, 334–336, 335

monitoring file changes with Windows services, 234–252

adding files to monitored folders, 250–252

from console applications, 244–245, 245

controlling from other applications, 246–250

and creating reports, 246–248, 250–251, 251

FileSystemWatcher component, 235, 236–240

installation warning, 245

installing in design mode, 243–244, 243–244

installing using InstallUtil.exe, 244–245, 245

interacting with, 248–250

launching, 245–246, 246

overview of, 234–235, 252  
 processing file changes, 235–236, 240–243  
 sending custom commands to, 247–250  
 ServiceController component, 235, 248–250  
 starting monitoring, 239  
 monitoring/launching external programs, **186–199**  
 overview of, 186–187  
 using Process class  
   example, 189–190  
   launching invisible processes, 194  
   launching processes, 192–193  
   monitoring launched processes, 192  
   overview of, 190–191  
   redirecting process IO, 196–199  
   retrieving invisible process output, 194–195  
   setting parameters at design time, 191, 191  
   waiting for launched process exit, 192–193, 193  
 in VB6, 187  
 using VB.NET Shell function, 187, 199  
 using Win32 ShellExecute function, 187–189, 196  
 multidimensional array sorting, **107–109**

---

## N

NameObjectCollectionBase, 172  
 namespace prefixes, creating, 125  
 navigating DataSets, **342–343**, 343  
 .NET classes. *See* classes  
 .NET code, embedding resource files in, 58, 111–113  
 .NET InstallUtil.exe, **244–245**, 245  
 .NET Unicode support, 282  
 newline, **87**  
 NewLine values shortcut, 264  
 normalization, database, 353  
 numbers, validating, **96–99**  
 numeric values in conversions, 157–158

---

## O

objects, *See also* collections  
   cloning, 79, 80  
   converting strings to, 167–169  
   creating Type instances of, 168–169  
   deserializing, 72, 76, 77  
   serializing, 71–73, 72–73, 75  
 opening files, 68–69  
 Option Strict On, importance of, 165  
 optional command line parameters, 212, 213  
 overloads for Merge method, **360–361**

## P

paging ASP.NET data in DataGrids, **412–421**, *See also* ASP.NET  
   adding paging controls, 419–420  
   using automatic paging, 414–416, 415–417, 420, 421  
   using custom paging, 416–420, 421, 421  
   customizing paging controls, 415–416  
   data-binding code, 420  
   versus not paging, 413, 414, 420  
   overview of, 412–413, 420–421  
   stored procedures in, 417–418  
   warning, 420  
 parsing/validating command lines, **212–234**, *See also* validating  
   command lines  
     defined, **212**  
     flag parameters, 212–213  
     optional parameters, 212, 213  
     splitting, 214  
   extending parsers, 220–234  
   generic guidelines for, 214–215  
   how parsers work, 215–217, 217  
   overview of, 212  
   parsing, defined, **143**  
   setting up parsers, 217–219, 219–220  
   warning, 216  
 Performance Monitor, **334–336**, 335  
 persisting object collections, 76–77  
 persisting objects, 73, 73, 75  
 Petroustos, Evangelos, 66, 81  
 PrimaryKey property, 362–363  
 Process class. *See* external programs  
 processing file changes, **235–236**, **240–243**, *See also* monitoring  
 PropertyManager class, 402

---

## R

reading/writing files. *See* file I/O; parsing  
 ReadOnlyCollectionBase, 172  
 ReadXml method, 387  
 Recordsets, 336–337, 342, 343, 346, *See also* ADO.NET DataSets  
 reference type conversions, **163–167**  
 RegisterClientScriptBlock method, 318  
 regular expressions, building custom, **89–99**  
   for finding dates in strings  
     alternation constructs and, 92  
     capturing groups, 95–96

- character classes and, 90, 93
- character escapes and, 93
- creating character classes, 93–96
- creating named groups, 95
- implementing rules, 90–93
- overview of, 89–90
- quantifiers and, 91
- word boundaries and, 93
- overview of, 86–87
- using Regex class
  - avoiding writing full class names, 87
  - compiling expression patterns, 88
  - expression symbols, 87, 89
  - matching individual characters, 87
  - newlines and, 87
  - overview of, 87
  - for validating numbers, 96–99
  - white space warning, 87
- relational databases. *See* databases
- relations between DataSet tables, **341–342**, 343
- Render method
  - overriding in message box output, 318–320
  - overriding in Web Forms output, 297–299
  - workaround, 318
- rendering design-time HTML, 312, 315, 320
- resource files (.resx), 277–280, 277–278, 280
- resource files (.xml), 58, 111–113
- ResourceManager objects, 292
- Response.Output.Writeline method, 264
- creating stored procedures, 389–390, 417–418
- drilling into connections, 328, 329
- reusing connection strings, 330
- testing connections, 328, 330
- viewing code behind, 329–330
- Shell function in VB.NET, 187, 199
- shell processes online resource, 187
- ShellExecute function in Win32 API, 187–189, 196
- showModalDialog method, 309, 314, 316, 317
- SoapFormatter class, 73, 73, 75
- sort capabilities, adding to classes
  - built-in QuickSort method, 100, 109
  - considering efficiency, 109
  - using IComparable interface, 100–104
  - using IComparer interface, 104–106
  - sorting
    - base type arrays, 100–101, 101
    - custom class arrays, 102–105
    - custom collections, 106–107
    - multidimensional arrays, 107–109
    - partial arrays, 105
    - typed arrays, 105–106
  - warning, 101
- splitting command lines, 214
- SQL Query Analyzer, 371
- SqlDataType/base type conversions, **162–163**
- SQLXML library, **369–379**, *See also* XML
  - defined, **369**
  - DiffGrams support, 370
  - using ExecuteXmlReader, 375–377, 377, 379
  - features, 369–370
  - generating XML with SQL statements
    - FOR XML AUTO clauses, 372–373
    - FOR XML EXPLICIT clauses, 373–374
    - FOR XML problem, 377
    - FOR XML RAW clauses, 370–372
    - FOR XML warning, 370
    - and getting into .NET applications, 375–377, 377
    - schema/data information, 374–375
    - using SqlXmlCommand object, 377–379, 379
    - turning FOR XML results into valid XML, 377–379, 379
    - XMLDATA option, 374–375
- stored procedures, creating, 389, 390, 417–418
- storing user and application data, **136–153**
  - in application configuration files, 151–152, 153
  - using custom file formats, 141–143, 153
  - in database tables, 153
  - guidelines for, 152–153
  - in INI files, 137–139, 138, 152
  - in Isolated Storage, 146–148, 153
  - overview of, 135–136
  - using Web services, 148–151, 153
  - in Windows Registry, 139–141, 153
  - in XML files, 143–145, 153

---

## S

- saving
  - DataSet changes to data sources, 345–346
  - DataSet changes to Web services, 408, 410, 412
  - merged data back to databases, 366
- Schema Designer, 347, **348–351**, 350
- schemas
  - creating schema-only DataSets, 386–387
  - generating with SQL, 374–375
  - merging with DataSet data, 364–366, 365
  - MissingSchemaAction values, 366
  - validating XML against, 121–123
  - XML Schema Web site, 351
  - XSD schema in creating DataSets, 350–351
  - XSD schema, creating in VS, 120–121
- serializing collections, 76–77
- serializing objects, 71–73, 72–73, 75
- Server Explorer GUI tool, *See also* ADO.NET database connections
  - accessing, 326–327
  - adding connection monitors, 336
  - creating connections, 327, 327, 330

streams. *See* file I/O

strings, converting

- to byte arrays, 160, 161
- byte arrays to, 161–162
- to Char arrays, 159
- to objects, 167–169
- value types to, 159

synchronizing DataSets with DiffGrams

- code behind, 409–412
- creating client applications, 406–407, 406
- creating Web services, 404–406, 404
- defined, **403**
- disconnected editing, 408, 409
- GetChanges method, 411
- minimizing data transfers, 403, 411
- overview of, 380
- saving changes, 408, 410, 412
- testing, 408, 408
- Web references and, 406–407, 406
- WriteXml method, 411

synchronizing DataSets with XML documents, **384–387**, 386

System handlers for custom tags, **256–258**

System.Xml namespace, 382

---

## T

tables. *See* ADO.NET DataSets; databases

tag handlers. *See* configuration handlers

testing

- CustomItemHandlers, 262–264, 263
- data-bound data-input forms, 391, 391, 394
- database connections, 328, 330
- DiffGrams, 408, 408
- Web Forms FocusManager control, 300–304, 305

text files, *See also* file I/O

- converting to XML, 73, 73, 75
- localizing in ASP.NET, 277–280, 277–278, 280, 292
- reading/writing, 67, 70, 71, 74

third normal form rule, **353**

times/dates, altering. *See* touch utilities

To String method, 159, 167

Toolbox, adding controls to, 299–300, 300, 321

touch utilities, building

- for altering file dates/times, 200–201
- defined, **200**
- using from console applications, 205–211, 205
- using from Windows forms, 211
- overview of, 200
- using TouchUtility class, 201–205

TreeView controls

- adding sorted lists, 107–109
- building XML-enabled controls, **48–64**
- customizing XML displays, 60–62, 60–61

- display capabilities needed, 51–52
- using DisplayRules, 53–55, 56–58, 60–62, 61–62
- embedded XML resource files and, 58
- extending TreeViewXml, 64
- finding XML node paths, 55–58
- how TreeViewXml works, 62–64
- iterating through XML nodes, 48–51, 51
- overview of, 48
- using XSLT option, 54–55, 56, 58, 60–62, 62

TypeConverter class, 165–166, 167–168, 169

typed array sorting, **105–106**

---

## U

Unicode

- in conversions, 157, 158, 160, 161
- defined, **157**
- .NET support for, 282

---

## V

validating data, *See also* command lines

- using regular expressions, 96–99
- with XmlValidatingReader, 120–123

value types, converting to strings, 159

ValueCollection, **179–186**

VB (Visual Basic)

- launching external programs in, 187
- ListBoxes, mimicking, 3–11, 8, 10–11
- versus VB.NET speed tests, 81–83

VB.NET

- compatibility syntax. *See* file I/O
- compatible random file access support, 67–68
- overview of, 81
- Shell function, 187, 199
- versus VB6 speed tests, 81–83

VBScript MsgBox method, 307–308, 309, 313, 314, 317

Visual Studio, creating XSD schema in, **120–121**

---

## W

Web Forms custom FocusManager control, *See also* ASP.NET

- adding properties, 296–297
- adding to Toolbox, 299–300, 300
- creating, 294–296
- overriding Render method, 297–299
- overview of, 294
- teting, 300–304, 305

Web references in DiffGrams, 406, **406–407**

- Web services, *See also* DiffGrams  
 saving DataSet changes to, 408, 410, 412  
 storing data using, 148–151, 153  
 in synchronizing DataSets/DiffGrams, 404–406, 404
- Web site addresses  
 Babel Fish, 267  
 BBC News Online, 267  
 DevX.com, 78, 78, 81  
 resource articles, 42, 58, 113, 187  
 Victor Garcia Aprea's Web log, 318  
 Win32 API INI functions, 38  
 XML Schema, 351
- Web sites, internationalizing in ASP.NET, **265–293**, *See also* ASP.NET  
 caching article content, 285–286  
 creating images dynamically, 287–290, 288, 291  
 improving culture matching, 292–293  
 language/culture issues, 266–267  
 letting users choose languages, 274–276, 276  
 localizing database content, 291  
 localizing dates, 290  
 localizing numbers/currencies, 290–291  
 localizing static images, 286–287, 287–288, 291–292  
 localizing text resources, 277–280, 277–278, 280, 292  
 managing content, 293  
 managing resource changes, 292  
 mapping content to classes, 267–274, 270, 273–274  
 overview of, 265  
 ResourceManager objects and, 292  
 specifying cultures, 281–284, 284
- web.config file, 254–259, 255, 262, 263, 264
- Weber, Phil, 81
- white space in expressions, 87
- Win32 API INI functions, **28–37**, 38
- Win32 API ShellExecute function, **187–189**, 196
- Windows forms, **2–64**  
 building XML-enabled TreeView controls  
   customizing XML displays, 60–62, 60–61  
   default controls, 58–59, 59  
   display capabilities needed, 51–52  
   using DisplayRules, 53–55, 56–58, 60–62, 61–62  
   embedded XML resource files and, 58  
   extending TreeViewXml, 64  
   finding XML node paths, 55–58  
   how TreeViewXml works, 62–64  
   iterating through XML nodes, 48–51, 51  
   overview of, 48  
   using XSLT option, 54–55, 56, 58, 60–62, 62
- drawing custom ListBoxes/ComboBoxes  
 displaying font names/styles, 25–27, 27  
 listing files/folders/icons, 17–23, 17  
 overview of, 16  
 with variable width/height items, 23–25, 23
- Listbox ItemData gone  
 assigning display methods to instances, 14–15, 15  
 class creators have control, 6–11, 8, 10–11  
 delegating control to consumers, 11–14  
 getting data back, 5–6  
 mimicking classic VB ListBoxes, 3–11, 8, 10–11  
 overview of, 2–3  
 populating ListBoxes, 3–5
- message boxes, 306–307, 307
- using touch utilities from, 211
- upgrading INI files to XML, *See also* INI  
 API INI functions, 28–37  
 comments, 47  
 extensibility, 46–47  
 INI sections/keys/values, 37–38  
 moving beyond COM Interop, 37  
 .NET configuration files and, 38–39  
 overview of, 28  
 retrieving values, 42–43  
 ubiquitous INI files, 37–38  
 updating/adding values, 44–45, 45  
 XML case sensitivity, 43–44  
 XML INI file structure, 39–42
- Windows Registry, storing data in, **139–141**, 153
- Windows services, monitoring file changes with, **234–252**  
 adding files to monitored folders, 250–252  
 from console applications, 244–245, 245  
 controlling from other applications, 246–250  
 controlling what to monitor, 236–239, 245, 248–250  
 and creating reports, 246–248, 250–251, 251  
 FileSystemWatcher component, 235, 236–240  
 installation warning, 245  
 installing in design mode, 243–244, 243–244  
 installing using InstallUtil.exe, 244–245, 245  
 interacting with, 248–250  
 launching, 245–246, 246  
 overview of, 234–235, 252  
 processing changes, 235–236, 240–243  
 sending custom commands to, 247–250  
 ServiceController component, 235, 248–250  
 starting monitoring, 239  
 warning, 245
- WriteXml method, 387, 411
- writing/reading files. *See* file I/O

---

## X

- XML, **28–64**, *See also* SQLXML  
 building XML-enabled TreeView controls  
 customizing XML displays, 60–62, 60–61  
 default controls, 58–59, 59  
 display capabilities needed, 51–52  
 using DisplayRules, 53–55, 56–58, 60–62, 61–62  
 embedded XML resource files and, 58

- extending `TreeViewXml`, 64
  - finding XML node paths, 55–58
  - how `TreeViewXml` works, 62–64
  - iterating through XML nodes, 48–51, 51
  - overview of, 48
  - using XSLT option, 54–55, 56, 58, 60–62, 62
- converting text files to, 73, 73, 75
- embedding as resources in .NET code, 58, 111–113
- storing data in, 143–145, 153
- `System.Xml` namespace, 382
- upgrading INI files to
  - API INI functions, 28–37
  - comments, 47
  - extensibility, 46–47
  - getting original files back, 47
  - moving beyond COM Interop, 37
  - .NET configuration files and, 38–39
  - overview of, 28
  - retrieving values, 42–43
  - ubiquitous INI files, 37–38
  - updating/adding values, 44–45, 45
  - XML case sensitivity, 43–44
  - XML INI file structure, 39–42
- XML classes, **109–134**
  - overview of, 109–110, 135
  - `XmlDataDocuments`, *See also* ADO.NET `DataSets`
    - creating `DataSets` from, 134–135
    - creating from `DataSets`, 132–134, 386
    - creating schema-only `DataSets` and, 386–387
    - defined, **132, 380, 384**
    - displaying as `DataSets` on `DataGrids`, 385, 386
    - overview of, 380
    - versus `ReadXml` method, 387
    - relational data and, 132, 387
    - using synchronized objects, 387
    - synchronizing, 384–387, 386
    - when to use, 132
    - versus `WriteXml` method, 387
  - `XmlTextReader/Writer` classes and, 385
    - in XPath data queries, 387
    - in XSLT data formatting, 132, 387
- `XmlDocuments`
  - default namespace prefixes and, 125
  - Document Object Model, 380–382, 382
  - loading XML into, 382–384, 383–384
  - specific XPath queries, 124–126
  - when to use, 124
  - `XmlNode` and, 382–383, 383–384
  - versus `XmlPathDocuments`, 127
  - versus `XmlTextReader`, 124, 126
  - `XmlTextReader/Writer` classes and, 384
  - versus `XmlTextWriter`, 124
- `XmlPathDocuments`
  - precompiling XPath queries, 128
  - specific XPath queries, 127
  - when to use, 127
  - versus `XmlDocuments`, 127
  - XPathNavigators, 128–129
  - XPathNodeIterators, 128–129
- `XmlTextReader`
  - in adding data to existing files, 117
  - `createEmbeddedFiles` method and, 110–113
  - `EmbeddedResourceReader` and, 112–113
  - error-trapping warning, 110
  - extracting/displaying specific values, 113–115
  - in modifying existing file data, 119
  - reading/displaying files, 110–113
  - retrieving stored XML data using, 145
  - when to use, 110
  - `XmlDataDocuments` and, 385
  - versus `XmlDocuments`, 124, 126
  - `XmlDocuments` and, 384
- `XmlTextWriter`
  - adding data to existing files, 117–118
  - constructing new documents, 115–117
  - modifying existing file data, 119–120
  - storing data in XML files using, 143–145
  - when to use, 115
  - `XmlDataDocuments` and, 385
  - versus `XmlDocuments`, 124
  - `XmlDocuments` and, 384
- `XmlValidatingReader`
  - creating schema before calling, 120–121, 122
  - handling validation errors, 123
  - validating XML against schema, 121–123
  - when to use, 121
- XslTransform stylesheets
  - customizing XML displays, 54–55, 56, 58, 60–62, 62
  - displaying XML sorted lists in HTML, 131–132
  - displaying XML in unsorted lists, 129–131
  - when to use, 129
  - `XmlDataDocuments` and, 132, 387
- XPath data queries
  - in `XmlDataDocuments`, 387
  - in `XmlDocuments`, 124–126
  - in `XmlPathDocuments`, 127, 128
- XSD (XML Schema Direct)
  - in creating typed `DataSets`, 350–351
  - creating in VS, 120–121