

# Index

## SYMBOLS & NUMERICS

% (percent sign) preceding custom format specifiers, 946  
**3D Computer Graphics (Watt, Allan H.), 762**

## A

**About dialogs, 279–281**

**abstraction, 412–414**

**accessibility**

- of classes, 423–424
- of constants, 163–164
- of events, 448–449
- of subroutines, 192–193
- of variables, 129, 138–139, 150–151

**Add Connection dialog, 307–308, 320–321**

**Add New Item dialog, 12, 13**

**Add Reference dialog, 14–15**

**Add Web Reference dialog, 15**

**AddHandler statement**

- catching events, 74–75, 450–451
- for custom events, 452
- making controls share events, 95
- setting event handlers at run time, 93–94

**Advanced Compiler Settings dialog, 112–113**

**aliases**

- with generics, 519
- listing for commands, 33
- for namespaces, 465–466, 519

**aliasing. See anti-aliasing**

**Alignment property (Pen class), 566–567**

**Alignment property (StringFormat class), 595, 642**

**Anchor property, 79–80, 85–88**

**anti-aliasing**

- aliasing defined, 542
- displaying text with and without, 531
- InterpolationMode property for, 544–545
- overview, 542–543
- SmoothingMode property for, 543
- speed considerations, 546
- TextRenderingHint property for, 543–544

**Application class or object**

- events, 703–705, 956
- methods (table), 702, 955
- My.Application namespace versus, 700
- properties, 700–702, 953–954

**application environment. See environment variables**

**application icons, 283**

**application resources. See resources**

**application settings. See configuring applications**

**arithmetic operators, 169–170, 781**

**Array class**

- Array.BinarySearch method, 485–486
- Array.Reverse method, 482
- Array.Sort method, 482–485
- IndexOf and LastIndexOf methods, 481–482
- multiple dimensions for arrays, 477
- nonzero lower bounds for arrays, 477–478
- normal arrays versus Array objects, 476–477, 479–481
- performance, 479–481
- resizing arrays, 478–479

**ArrayList class, 486–488**

**arrays. See also Array class**

- ArrayList class for resizable arrays, 486–488
- changing dimensions of, 478
- circular, in queues, 505
- creating with variable declarations, 476
- declared ByVal and ByRef, 195–196
- IndexOf and LastIndexOf methods, 481–482
- initializing, 148–150
- normal arrays versus Array objects, 476–477
- of objects, initializing, 150
- parameter arrays for subroutines, 198–200
- performance, 479–481
- structures versus classes and, 433–434

**assemblies, defined, 107**

**Assembly Information dialog, 106–107**

**Assert method, 235**

**assignment operators, 175–176, 784**

**attribute\_list clause**

- for classes, 422
- for events, 448

# attribute\_list clause (continued)

---

## attribute\_list clause (continued)

- for subroutines, 188–190
  - for variables, 138
- attributes for custom controls, assigning, 368–369**
- Audio object (My.Computer), 963**
- AutoRedraw implementation, 620–624**
- Autos window, 28**

## B

**BackgroundWorker component, 805–806**

### Bézier curves

- defined, 523
- DrawBezier method for, 534–535
- DrawBeziers method for, 535–536

### binary-file access, 723

**BinaryReader class, 711–712, 973–978**

**BinaryWriter class, 711, 713, 978**

### BindingNavigator component

- attaching BindingSource component to, 814
- automatic creation for databases, 313
- DataGridView control for, 814
- overview, 806

**BindingSource component, 807, 814**

### Bitmap class

- GetPixel method, 625–626, 630
- implementing AutoRedraw, 620–624
- loading bitmaps, 618–619
- LockBits method, 627
- methods (table), 617–618, 936–937
- overview, 617–618, 632–633
- pixel-by-pixel operations, 625–630
- saving bitmaps, 619–620
- SetPixel method, 625–626, 630

**BitmapBytesRGB24 class, 629–630**

### bitwise operators, 174, 783

### blocks

- defined, 154
- scope of variables, 154–155

### bookmarks in code editor, 53

**bounds\_list clause for variables, 146–147**

**Breakpoint Condition dialog, 30–31**

**Breakpoint Hit Count dialog, 31**

### breakpoints

- code editor commands for, 53
- common debugging strategy, 30
- counting hits for, 31
- defined, 30
- setting actions for hits, 31–32
- setting conditions for, 30–31
- setting in property procedures, 160

### Breakpoints window

- opening, 26
- overview, 30–32

### Brush class

- derived classes, 572, 934
- garbage collection issues, 590–591
- HatchBrush derived class, 572, 574–575
- LinearGradientBrush derived class, 572, 575–580

- overview, 522, 572
- PathGradientBrush derived class, 572, 580–585
- Pen class versus, 563
- SolidBrush derived class, 572
- TextureBrush derived class, 572, 572–575

**BufferedStream class, 711**

### bugs. See also debugging; error handling

- catching, 234–235
- defined, 233
- first computer bug, 234
- unplanned conditions versus, 233–234

### Build menu

- commands, 20–22
- Configuration Manager command, 20–22
- overview, 18, 19

### building

- custom components, 377–379
- custom controls, approaches, 361–362
- custom controls from scratch, 362, 376–377
- data sources, 312–314
- DataSet component, 332–334
- MDI applications, 292–293
- Metafile, 630
- MRU lists, 296–300
- reports, 656–666
- RowSortingListView derived control, 370

**Business Objects Web site, 655**

**Button control, 270, 807**

**Button Editor dialog, 42, 43**

### buttons

- Button control for, 807
- for navigation, 806
- with PrintPreviewDialog component, 868–869
- RadioButton control for, 871–872

### ByRef keyword

- array parameters and, 195–196
- overview, 157, 158–159
- parameter passing and, 436–437
- in subroutine declarations, 194–196

### ByVal keyword

- array parameters and, 195–196
- overview, 157–158
- parameter passing and, 436–437
- in subroutine declarations, 193–194, 195–196

## C

**calendars, MonthCalendar control for, 267, 847–852**

**Call Stack window, 28–29**

**CancellationPending property  
(BackgroundWorker component), 805**

**Catch statement, 239, 240–241**

**catching events, 74–75, 450–451**

**CausesValidation property, 95, 97**

**centering printouts, 647–649**

**Change Data Source dialog, 307**

**Change Group Options dialog, 668–669**

### characters. See also operators

- data type (table), 132, 133, 777–778
- date and time format specifiers, 943–946

- with Like operator, 171–172, 783
- literal type (table), 133, 778
- for MaskedTextBox control Mask property, 842–843
- numeric format specifiers, 947–949
- CheckBox control, 266, 807–808**
- CheckedListBox control**
  - displaying text in Output window, 810
  - Items.Add method, 809–810
  - overview, 266, 808–810
- Choose statement**
  - overview, 214–215, 791
  - Select Case statement versus, 214–215
- Choose Toolbox Items dialog, 50, 51**
- Class statement**
  - code regions defined by, 109
  - top-level nodes defined by, 108
- classes. See also collection classes; generics; inheritance; specific classes**
  - accessibility, 423–424
  - adding and modifying features, 415–416
  - attribute\_list clause, 422
  - code regions, 109–110
  - command object classes, 330
  - constructors, 438–440, 922
  - for custom controls, 380
  - for custom exceptions, 248–249, 796
  - data types for parameters, 925
  - declaring classes, 421–431, 921–922
  - declaring custom events, 451–455
  - declaring events normally, 448–449
  - declaring methods, properties, and constants within, 446–447
  - defined, 407
  - derived from generics, 519
  - DictionaryBase class, 497–499
  - Dispose method, 444–446
  - drawing classes, 930–938
  - encapsulation by, 409–410, 446–447
  - exception classes, finding, 247–248
  - exception classes (table), 796–798, 939–941
  - Finalize method, 442–444
  - garbage collection, 442–446
  - generic collection classes, 507–508
  - heap and, 434
  - Implements keyword, 427–431
  - inheritance clause, 425–426
  - Inherits statement, 427
  - instantiation, 409, 438–440
  - memory requirements, 433–434
  - name collisions and, 459
  - namespaces versus, 469–470
  - object assignment by, 435–436
  - Of type\_list clause, 426
  - overloading, 418–420
  - overview, 407–409
  - parameter passing by, 436–437
  - Partial keyword, 422–423
  - performance and, 434
  - public interface, 409–410
  - as reference types, 432–433
  - sections, 108–109
  - Shadows keyword, 424–425
  - shared methods in, 456–458
  - shared variables in, 455–456
  - stream classes (table), 708, 975–976
  - structures versus, 431, 432–437, 458, 922
  - syntax, 421
  - in System.Drawing namespace (table), 927–928
  - in System.Drawing.Drawing2D namespace (table), 525–526, 928–929
  - in System.Drawing.Imaging namespace (table), 528, 929
  - in System.Drawing.Printing namespace (table), 532, 930
  - in System.Drawing.Text namespace (table), 529, 930
  - unit scope of, 446–447
  - unmanaged resources and, 442–446
- classic error handling**
  - error-handling mode for, 253–254
  - exiting error-handling code, 799
  - On Error GoTo line statement, 250–251, 799
  - On Error GoTo -1 statement, 252–253, 799
  - On Error GoTo 0 statement, 252, 799
  - On Error Resume Next statement, 251–252, 799
  - structured error handling versus, 254–256
- client area of controls, 89**
- clipboard**
  - Clipboard object methods (table), 404–405, 963–964
  - copy and paste using, 404–405
  - copying data to, 402–403
  - overview, 385
  - retrieving data from, 403
- Clipboard object (My.Computer), 404–405, 963–964**
- Clock object (My.Computer), 964**
- cmd command, 32**
- code editor**
  - bookmarks, 53
  - breakpoint commands, 53
  - code coloring and highlighting, 57–59
  - code snippets, 59–64
  - creating event handlers, 91–92
  - IntelliSense, 56–57
  - margin icons, 52–53
  - outlining, 53–55
  - overview, 51–52
  - at run time, 54
  - tooltips, 55–56, 58
- code file structure**
  - code regions, 109–110
  - conditional compilation, 110–118
  - line continuation, 126
  - line joining, 126–127
  - line labels, 127
  - namespaces, 118–120
  - overview, 108–109, 127–128
  - sections, 108
  - typographic code elements, 120–127
  - Visual Basic comments, 121
  - XML comments, 122–125
- code reuse**
  - generic classes and, 512
  - inheritance and, 411–412

## code snippets

Code Snippets Manager, 59, 63–64  
creating, 62–63  
defined, 59  
using, 60–62

**Code Snippets Manager, 59, 63–64**

**code window. See code editor**

**Collection class, 475, 476**

### collection classes

Array class, 475, 476–486  
ArrayList class, 486–488  
Collection class, 475, 476  
collection defined, 475  
CollectionBase class, 488–491  
CollectionsUtil class, 501–502  
dictionaries, 493–501  
generics, 507–508  
HashTable class, 496–497  
HybridDictionary class, 497  
ListDictionary class, 494–496, 500  
NameValueCollection class, 491–493  
overview, 475–476, 508–509  
queues, 501, 504–507  
read-only strongly typed collections, 491  
ReadOnlyCollectionBase class, 491  
SortedList class, 500–501  
stacks, 501–504  
StringCollection class, 488  
StringDictionary class, 500  
strongly typed collections, 488–491

**collection properties controls, 80–83, 84**

**CollectionBase class, 488–491**

**CollectionsUtil class, 501–502**

**ColorDialog component, 810–811**

### colors

in code editor, 57–58  
ColorDialog component for, 810–811

**ColorScroller composite control, 374–375**

**ComboBox control**

binding, 356–359  
overview, 267, 812–813

### command objects

classes, 330  
creating for data adapters, 326–327  
defined, 317  
methods, 331–332  
properties, 330–331

**Command window, 32–33**

### comments

normal, 121  
XML, 122–125

### comparison operators

characters with Like, 171–172, 782  
Is, 171  
IsNot, 171  
Like, 171–172, 782–783  
Like patterns, 172, 783  
overview, 170–172  
table summarizing, 171, 782  
TypeOf, 171

## Compile tab (Properties Page)

Option Compare, 18  
Option Explicit, 16, 18, 776  
Option Strict, 17–18, 776

## compiling

conditional compilation, 110–118  
Option Compare compiler option, 18  
Option Explicit compiler option, 16, 18, 151–153, 776  
Option Strict compiler option, 17–18, 153, 776

**Component class. See also components**

for custom components, 380

inheritance from, 801

**ComponentResourceManager class, 698–700**

**components. See also Component class; controls; Tool-**

**box; specific components**

adding at design time, 68–70  
controls versus, 67, 68  
custom, 362, 377–379, 382  
defined, 67, 377  
information provided for controls by, 69  
inheritance, 801  
listed by purpose, 803–804

**composite controls, 361, 374–376**

**composite formatting, 950**

**compound properties of controls, 77–78**

**CompoundArray property (Pen class), 567–568**

## concatenation

operators, 170, 782  
StringBuilder class for, 176–178

## conditional compilation

CONFIG constant, 113, 116–117  
DEBUG constant, 113, 114–116  
DEBUG\_LEVEL constant, 117–118  
overview, 110–111  
predefined constants, 113–118  
setting constants in code, 111–112  
setting constants with project settings, 112–113  
TARGET constant, 113  
TRACE constant, 113, 116  
VBC\_VER constant, 113

## CONFIG constant

defined, 113  
overview, 116–117

## configuration files

app.config file example, 687–688  
assigning dynamic properties to, 685–687  
My.Settings namespace properties and methods for, 688–689  
overview, 685  
uninstalling programs and, 683

## Configuration Manager dialog

creating new configurations, 21  
Debug versus Release configurations, 21  
excluding projects from builds, 22  
overview, 20–22

**configuring applications. See also resources**

configuration files for, 685–689  
environment variables, 677–681  
hidden files for, 104, 105

- overview, 675, 705
- Registry for, 681–685
- settings versus resources, 104
- connection objects**
  - adding connections, 320–321
  - ConnectionString property, 319–320
  - databases support by, 319
  - defined, 317, 318
  - further information, 318–319
  - methods (table), 322
  - performance and, 319
  - properties (table), 319
  - types of, 318–319
- connection wizard, 659–660**
- ConnectionString property, 319–320
- console streams, 980**
- Const keyword, 163
- constants. See also specific constants**
  - accessibility, 163–164
  - for conditional compilation, 111–118
  - declaring, 163–164, 446–447
  - initialization\_expression clause, 164
  - Option Strict compiler option and, 164
- constrained types with generics, 516–518**
- constraints (database)**
  - DataSet support for, 345
  - defined, 345
  - ForeignKeyConstraint, 345–346
  - UniqueConstraint, 345, 346–347
- constructors**
  - for classes, 438–440, 922
  - generic, 514
  - Pen class (table), 564
  - for stacks, overloaded, 503–504
  - for structures, 440–441, 922
- container controls**
  - adding controls to, 73
  - dragging, 72
  - FlowLayoutPanel control, 265, 820–821, 889
  - GroupBox control, 265, 826
  - list of, 803
  - overview, 265–266
  - Panel control, 265–266, 859–860
  - SplitContainer control, 266, 879–881
  - TabControl control, 266, 882–886
  - TableLayoutPanel control, 265, 821, 886–889
- container types, 421. See also classes; structures**
- ContextMenuStrip component, 270, 813–814
- Continue statement
  - For loops, 218, 221, 228
  - While loops, 226–227, 228
- continuing lines of code, 126**
- control array events, 94–95**
- Control class. **See also controls**
  - for custom controls, 380
  - customizing in executable projects, 382
  - events (table), 769–771
  - inheritance from, 84–85, 801
  - inherited in executable projects, 382
  - methods (table), 766–768
  - properties (table), 763–766
- Control Panel (Windows XP), 677–678**
- control statements. See also specific kinds**
  - decision statements, 205–215, 789–791
  - defined, 205, 789
  - GoTo, 228–231, 793
  - looping statements, 216–228, 791–793
- controls. See also components; custom controls; Toolbox; specific controls**
  - AddHandler statement for, 74–75, 93–94, 95
  - adding to containers, 73
  - binding (complex strategy), 356–359
  - binding (simple strategy), 351–352
  - catching events, 74–75
  - changing events at design time, 94
  - choosing, 264–272
  - client area, 89
  - collection properties, 80–83, 84
  - components versus, 67, 68
  - compound properties, 77–78
  - container controls, 72, 73, 265–266
  - control array events, 94–95
  - copying, 71–72
  - creating at design time, 70–73
  - creating at run time, 73–75
  - creating event handlers at design time, 91–92
  - defined, 67
  - display area, 89
  - for displaying data, 268–269
  - for displaying dialogs, 271–272
  - for displaying graphics, 271
  - for entering data, 267–268
  - events, 90–100, 769–771
  - forms as, 275
  - information provided by components, 69
  - inheritance, 84–85, 801
  - for initiating action, 270–271
  - listed by purpose, 803–804
  - for making selections, 266–267
  - methods, 90, 766–768
  - nonclient area, 89
  - overview, 100–101
  - position and size properties, 89–90
  - properties defined, 75
  - properties (table), 763–766
  - for providing feedback, 269–270
  - raising events, 90
  - read-only properties, 84
  - RemoveHandler statement for, 93–94
  - restricted properties, 78–80
  - selecting from Toolbox, 72–73
  - setting event handlers at run time, 93–94
  - setting properties at design time, 76–83
  - setting properties at run time, 83–84
  - specialized property editors for, 78, 79–80
  - for supporting other controls, 272
  - third-party controls, 272–273
  - validation events, 95–100
  - Windows forms controls (table), 259–264
  - WithEvents variables for, 74, 92–93
- Convert dialog, 6, 7**

## **converting applications from Visual Basic 6 to Visual Basic 2005, 6, 7**

### **converting data types**

- functions (table), 135–136, 778–779
- narrowing conversions, 134–137
- Option Strict compiler option and, 17, 153
- parsing methods, 137
- widening conversions, 137

### **copying**

- controls, 71–72
- copy and paste using clipboard, 404–405
- data to clipboard, 402–403
- dragging items between ListBoxes, 392–393

### **Create GUID dialog, 37, 38**

### **cryptography, 747–751**

### **CryptoStream object, 748**

### **Crystal Reports for Visual Studio 2005**

- building a report, 656–666
  - Crystal Reports Gallery, 657, 658
  - CrystalReportViewer control, 666–667, 671–673
  - further information, 655
  - Report object, 655–656
  - Standard Report Creation Wizard, 657–665
- ### **CrystalReportViewer control**
- external reports with, 671–672
  - manipulating output programmatically, 672–673
  - ReportDocument object and, 672–673
  - ReportSource property, 671
  - viewing reports, 656, 666–667

### **CurrencyManager object**

- methods (table), 353
- overview, 352
- program demonstrating, 353–355
- properties (table), 352–353

### **Cursor property of forms, 281**

### **Cursors class, 281–282**

### **cursors for forms, 281–282**

### **curves**

- Bézier, 523, 534–536
- DrawBezier method for, 534–535
- DrawBeziers method for, 535–536
- DrawClosedCurve method for, 535–538

### **custom components**

- building, 377–379
- in executable projects, 382
- overview, 362

### **custom controls**

- approaches to building, 361–362
- assigning attributes, 368–369
- building from scratch, 362, 376–377
- ColorScroller composite control, 374–375
- composite controls, 361, 374–376
- derived controls, 361, 370–374
- in executable projects, 380–382
- implementing properties, methods, and events, 366–367
- inheritance for, 380
- invisible controls, 379–380
- making the control project, 362
- managing design time and run time, 369
- overview, 361–362, 382–383
- RowSortingListView derived control, 370–374
- setting the Toolbox icon, 363

SimpleSmiley custom control, 376–377

- steps for building, 362
- test project for, 364–365
- testing, 365–366
- UserControl Test Container for, 363–364
- uses for, 361

### **custom dialogs**

- About dialogs, 279–281
- ColorDialog component for, 810–811
- FolderBrowserDialog component for, 821–823
- FontDialog component for, 823–826
- forms as, 302–303
- list of components for, 804
- OpenFileDialog component for, 855–857
- SaveFileDialog component for, 877–878
- wizards, 303–304

### **custom events, 451–455**

### **custom exception classes, 248–249, 796**

### **custom format specifiers, 944–946**

### **custom menus**

- ContextMenuStrip component for, 270, 813–814
- MenuStrip control for, 270, 845–847
- MRU lists, 296–302
- tools for, 804
- Window menu for MDI child forms, 292, 293

### **custom stream classes, 716**

### **custom toolbars**

- StatusStrip control for, 270, 881–882
- tools for, 804
- ToolStrip control for, 270–271, 891–893
- ToolStripContainer control for, 893–894

### **CustomEndCap property (Pen class), 567–568**

### **Customize command (Tools menu), 40**

### **Customize dialog**

- Commands tab, 41
- Keyboard button, 43–44
- opening, 40
- Rearrange Commands button, 43
- right-click menu for, 42–43
- Toolbars tab, 40

### **customizing reports, 667–671**

### **CustomStartCap property (Pen class), 567–568**

## **D**

### **Data Adapter Configuration Wizard, 327–330**

### **data adapters**

- associating command builder object with, 326–327
- creating at design time, 327–330
- creating command objects, 326–327
- defined, 317, 326
- overview, 317, 318
- in Relation editor, 312

### **data binding**

- for ComboBox control, 356–359
- complex, 356–359
- for ListBox control, 359
- simple, 351–352

### **data containers**

- DataSet object, 317–318, 332–336
- defined, 317

### **Data menu, 33–34**

**Data Source Configuration Wizard**

- changing the default data source, 307
- choosing a data source, 305–306
- choosing database objects, 310–311
- choosing or creating a data connection, 306–309
- opening, 305
- saving the connection string, 308–310

**Data Sources window, 312–314****data types**

- characters (table), 132, 777–778
- characters, using, 131–134
- for constants, 164
- constrained, with generics, 516–518
- conversion functions (table), 135–136, 778–779
- for drag and drop, 386
- elementary (table), 130–131
- enumerated types, 161–163, 776, 950–951
- for generic class parameters, 925
- integer, with explicit sizes, 131
- literal type characters (table), 133, 778
- literal types, using, 133–134
- multiple declarations in single statement and, 151
- multiple, with generics, 514–516
- narrowing conversions, 134–137
- parsing methods, 137, 779
- performance and, 131
- table summarizing, 776–777
- variable behavior determined by, 129
- widening conversions, 137

**Database Explorer window, 320****database programming. See also reporting**

- automatically created objects, 314–316
- building a data source, 312–314
- command objects, 317, 326–327, 330–332
- complexity of, 305
- connecting to data automatically, 305–314
- connection objects, 317, 318–322
- constraints, 345–347
- CurrencyManager object, 352–355
- data adapters, 312, 317, 318, 326–330
- data binding, complex, 356–359
- data binding, simple, 351–352
- data containers, 317
- data overview, 317–318
- Data Source Configuration Wizard, 305–311
- DataColumn object, 341–343
- DataRelation object, 343–345
- DataRow object, 339–341
- DataRowView object, 350–351
- DataSet object, 332–336
- DataTable object, 336–339
- DataGridView object, 347–350
- objects for database access, 316–317
- overview, 359–360
- Relation editor, 312
- Schema Editor, 311–312
- transaction objects, 322–326

**DataRow object**

- examples, 342–343
- overview, 341
- properties (table), 341–342

**DataConnector control, 272****DataFormats class, 393–394****DataGridView control, 339–340****DataGridView control**

- automatic creation for databases, 313
- overview, 268, 814

**DataNavigator control, 272****DataObject class, 399–400****DataRelation object**

- nested tables using, 344–345
- overview, 343–344
- properties (table), 344

**DataRow object**

- DataGridView control and, 339–340
- DataRowView object and, 350
- methods (table), 340–341
- overview, 339
- properties (table), 339

**DataRowView object, 350–351****DataSet component**

- building and initializing, 332–334
- methods, 335–336
- overview, 317–318, 814
- properties, 334–335
- uses for, 332

**DataTable object**

- events, 339
- methods, 338
- overview, 336
- properties, 337–338

**DataGridView object**

- binding to controls, 347
- DataRowView object and, 350
- example, 347–349
- methods (table), 350
- overview, 347
- properties (table), 349–350
- sorting and, 350

**Date data type, operations with, 178–181, 785****Date object format specifiers, 943–946****dates**

- custom format specifiers, 944–946
- Date data type operations, 178–181, 785
- DateTimePicker control for, 814–816
- standard format specifiers, 943–944, 945, 946
- TimeSpan data type operations, 179–180, 785

**DateTimePicker control**

- Format property, 815–816
- overview, 267, 814–816
- ShowUpDown property, 814–815

**Debug configurations, Release configurations versus, 21****DEBUG constant**

- Debug.Assert method, 114–116
- Debug.Fail method, 116
- defined, 113
- indentation properties, 116
- output methods, 116
- overview, 114
- for validating data, 235

**Debug menu**

- commands, 23–26
- overview, 22–23
- Windows submenu, 26–30

# Debug object

---

## Debug object, 235

### debugging. *See also conditional compilation; error handling*

- adjusting level of diagnostic output, 117–118
- custom controls, 363–364
- Debug configurations, 21
- Debug menu, 22–30
- overview, 257
- unplanned conditions versus bugs, 233–234

## DEBUG\_LEVEL constant, 117–118

## Debug.Print statement, 32

### decision statements

- Choose, 214–215, 791
- defined, 205, 789
- If, 212–214, 791
- multiline If Then, 207–208, 790
- Select Case, 208–212, 790
- single-line If Then, 205–206, 789–790

### declaring. *See also variable declarations*

- classes, 421–431, 921–922
- constants, 163–164, 446–447
- custom events, 451–455
- enumerated types, 161
- events, 448–449, 922–923
- functions, 202–203, 787–788
- generics, 512–513, 925
- methods within classes, 446–447
- operator overloading, 181–182
- parameters, 157–159
- properties within classes, 446–447
- property procedures, 160, 203, 788
- structures, 431, 922
- subroutines, 187–202, 787

### deferred validation, 98–100

### delegates

- for custom events, 452
- defined, 165
- overview, 165–166

## DeleteSetting method, 683

### derived classes, from generics, 519

### derived controls

- hiding parent features, 373–374
- overview, 361, 370–372
- shadowing parent features, 372–373

### design time

- adding components at, 68–70
- changing events at, 94
- control properties at, 76–83
- creating controls at, 70–73
- creating data adapters at, 327–330
- creating event handlers at, 91–92
- managing custom controls for, 369
- methods and, 90

### dialogs, custom. *See custom dialogs*

### dictionaries

- defined, 493
- DictionaryBase class, 497–499
- HashTable class, 496–497
- HybridDictionary class, 497
- ListDictionary class, 494–496, 500
- properties and methods, 493–494
- SortedList class, 500–501

StringDictionary class, 500

strongly typed dictionaries, 497–500

DictionaryBase class, 497–499

### Dim keyword

- for constants, 163
- for variables, 143

Directory class, 723–724, 985–986

DirectoryEntry component, 816

DirectoryInfo class, 727–729, 989–990

directory-manipulation methods, 719–720, 982–983.

*See also file systems*

DirectorySearcher component, 816

### Direct3D

- adding references to libraries for, 756
- defined, 756
- further information, 762
- program demonstrating, 756–762

### DirectX, 756

### display area of controls, 89

Dispose method, 444–446, 590–591, 731

### Do Loop statement

- Continue Do statement, 226–227, 228
- enumerators, 223–224
- Exit Do statement, 226, 227–228
- iterators, 224–225
- overview, 225–227
- repeating forever, 225, 792
- testing condition after execution, 226, 792
- testing condition before execution, 225, 792
- While End statement and, 793

Dock property, 85, 88–89

DoDragDrop method, 387, 389, 392, 394, 396

DomainUpDown control, 267, 816–817

### drag and drop

- accepting dropped files, 395–396
- changing format names, 399–400
- checking for textual format, 388–389
- drag source defined, 385
- drag source events, 387
- dragging multiple data formats, 400–402
- dragging objects, 396–399
- dragging within an application, 394–395
- drop target defined, 385
- drop target events, 386–387
- learning data types available, 393–394
- moving and copying between ListBoxes, 392–393
- moving between ListBoxes, 389–392
- overview, 405–406
- simple example, 387–388
- steps for, 385

DrawBezier method, 534–535

DrawBeziers method, 535–536

DrawClosedCurve method, 535–538

drawing. *See also drawing classes; Graphics class;*

### image processing

- advanced transformations, 551–556
- anti-aliasing, 531, 542–546
- basic steps, 522–523
- Bézier curves, 534–536
- closed curves, 536–537
- curves, 534–538

events, 558–561  
 filling areas, 539  
 fitting pictures to the page for printing, 649–651  
 garbage collection issues, 590–591  
 GraphicsPath objects, 585–590  
 namespaces, 523–532, 927–930  
 outlines, 538–539  
 overview, 521–523  
 Paint event handler for, 522  
 printing and, simplifying, 651–654  
 saving and restoring graphics states, 556–558  
 text, 593–595  
 transformation basics, 546–551  
**drawing classes. See also drawing; Graphics class; specific classes**  
 Bitmap class, 617–630, 937–938  
 Brush classes, 572–585, 934  
 GraphicsPath class, 585–590, 935  
 Image class, 615–617, 936–937  
 Metafile class, 630–633, 938  
 namespaces, 523–532, 927–930  
 Pen class, 563–571, 933–934  
 StringFormat class, 595–605, 936  
 DrawPath **method**, 538–539  
 DrawString **method**, 593–595  
 DriveInfo **class**, 726–727, 988–989  
**drop target. See also drag and drop**  
 defined, 385  
 events, 386–387

## E

**Edit menu, 8–10**  
**editors. See also code editor**  
 for icons, 282–283  
 Relation editor, 312  
 Schema Editor, 311–312  
 specialized property editors for controls, 79–80  
**embedded resources, 694–695**  
**encapsulation**  
 breaking, confusion from, 446–447  
 by classes, 409–410  
 by code libraries, 410  
**End Region statement, comments for, 109–110**  
**enumerated types**  
 declaring, 161  
 defined, 161  
 formatting, 950–951  
 overview, 161–163  
 variable declaration for, 161, 776  
**enumerations**  
 HatchStyle, 527, 576  
 LineCap, 527  
 LineJoin, 527  
 in System.Drawing.Drawing2D namespace (table), 526, 928–929  
 in System.Drawing.Text namespace (table), 530  
**enumerators, 223–224**  
**Environ function, 679**  
**environment variables**  
 Environment Variables dialog (Windows XP), 678–679  
 modifying with System.Environment object, 679–681  
 retrieving with Environ function, 679  
**error handling. See also debugging; exceptions**  
 bugs versus unplanned conditions, 233–234  
 catching bugs, 234–235  
 catching unplanned conditions, 235–238  
 classic, 250–254, 799  
 custom exceptions for, 248–249, 796  
 Err object, 256–257  
 error-handling mode, 253–254  
 exception classes, 796–798  
 Exception objects, 241–243  
 exiting error-handling code, 799  
 global exception handling, 238–239  
 need for, 233  
 On Error statements, 250–253, 799  
 overview, 257–258  
 StackTrace objects for, 243–246  
 structured, 239–249, 795  
 structured versus classic, 254–256  
 throwing exceptions, 246–248, 795  
 Try block for, 239–241, 795  
**Error List window, 4–5**  
**ErrorProvider component, 269, 817–818**  
**event handlers. See also drag and drop**  
 creating at design time, 91–92  
 for drawing events, 558–561  
 for global exception handling, 238–239  
 for MDI forms, 293–295  
 overview, 91  
 setting at run time, 93–94  
 for validation events, 95–100  
 WithEvents keyword for, 74, 92–93  
**Event statement, 448–449**  
**EventLog component, 818–820**  
**events. See also events, tables summarizing**  
 accessibility, 448–449  
 AddHandler statement, 74–75, 93–94, 95, 450–451  
 assigning attributes for custom controls, 368  
 attribute\_list clause, 448  
 catching, 74–75, 450–451  
 changing at design time, 94  
 control array events, 94–95  
 creating handlers at design time, 91–92  
 custom, 451–455  
 DataTable object (table), 339  
 declaring custom events, 451–455  
 declaring normally, 448–449, 922–923  
 defined, 407  
 drag-and-drop, 386–387  
 drawing events, 558–561  
 Form class (property-changed events), 918–919  
 implementing for custom controls, 366–367  
 Implements keyword, 449  
 MaskedTextBox control (table), 845  
 MDI, 293–295  
 mouse events sequence, 772  
 move events sequence, 773  
 overview, 90–91  
 parameters clause, 449

### events (continued)

PrintDocument component, 636–637  
properties and methods versus, 448  
raising, 90, 450, 452–453, 923  
RemoveHandler statement for, 93–94  
resize events sequence, 772–773  
sequences, 772–773  
setting handlers at run time, 93–94  
Shadows keyword, 449  
table summarizing, 769–771  
validation events, 95–100  
WithEvents keyword for, 74, 92–93, 450

### events, tables summarizing

Application class, 704, 956  
Control class, 769–771  
drag source events, 387  
drop target events, 386  
FileSystemWatcher class, 992  
Form class, 914–918  
Form class (property-changed events), 919  
MaskedTextBox control, 845  
My.Application, 957–962  
Network object (My.Computer), 968  
SerialPort class, 970  
SplitContainer control, 881

### Exception class, 242

### Exception objects, 241–243

### exceptions. See also error handling

classes, finding, 247–248  
classes (table), 796–798, 939–941  
custom classes, 248–249, 796  
Exception objects for, 241–243  
global exception handling, 238–239  
throwing, 246–248, 795  
Try block for, 239–241, 795

### Exceptions dialog, 23–24

### Exit statement

For loops, 218, 221, 227–228  
While loops, 227–228

### Exit Sub statement, 230–231

### exiting error-handling code, 799

### Export Template Wizard, 7–8

### Extensible Markup Language. See XML

### external reports, 671–672

## F

### FIFO lists. See queues

### File class, 724–726, 986–988

### File menu, 5–8

### file systems. See also files

binary-file access, 723  
file methods (table), 718, 981–982  
file-system methods (table), 719–720, 982–983  
My.Computer.FileSystem object for, 734–736, 965–966, 995–998  
.NET Framework classes for, 723–734  
random-file access, 720–722  
sequential-file access, 720  
System.IO Framework namespace for, 983–995

Visual Basic commands for manipulating, 718–723

Visual Basic methods for, 981–983

### FileInfo class, 729–730, 991–992

### files. See also file systems

accessing namespaces inside, 120, 462–463  
cryptography, 747–751  
hidden, 103–107  
methods for manipulating (table), 718, 982–983  
separate, for related pieces of code, 110

### FileStream class, 709–710

### FileSystem class (System.IO), 983–985

### FileSystem object (My.Computer)

methods (table), 735–736, 965–966, 992–997  
properties (table), 734–735, 965–966, 995  
SpecialDirectoriesProxy object, 737, 997–998

### FileSystemInfo class, 731

### FileSystemNeeded property, 428

### FileSystemWatcher class

Dispose method, 731  
events (table), 992  
methods (table), 992  
overview, 731, 992  
properties (table), 731, 992  
using, 732–733  
WaitForChanged method, 732

### FileSystemWatcher component, 820

### filling areas of drawings, 539

### Finalize method, 442–444

### firing events. See raising events

### fitting pictures to the page for printing, 649–651

### FlowLayoutPanel control

overview, 265, 820–821  
TableLayoutPanel control versus, 821, 889

### FolderBrowserDialog component, 821–823

### Font property, 78

### FontDialog component, 823–826

### fonts. See also text

listing system fonts, 529–530  
measuring width and height of text, 605–607  
metrics, 609–612

### For Each statement

Continue For statement, 221, 228  
enumerators versus, 224  
Exit For statement, 221, 227  
interface for groups with, 220  
overview, 220–225  
syntax, 220, 291

### For Next statement

Continue For statement, 218, 228  
Exit For statement, 218, 227–228  
noninteger loops, 219  
overview, 216–219  
syntax, 216, 291

### ForeignKeyConstraint, 345–346

### Form class

events (table), 914–918  
inheritance trail for, 275  
methods (table), 911–914  
overview, 905

properties (table), 905–911  
property-changed events, 918–919

#### Form Designer, 4

#### Form object

Opacity property, 275–277, 279  
TransparencyKey property, 277–279

#### Format menu, 34–36

Format **property** (DateTimePicker **control**), **815–816**

#### format specifiers

composite formatting, 950  
date and time, custom, 944–946  
date and time, standard, 943–944, 945, 946  
enumerated type formatting, 950–951  
numeric, custom, 948–949  
numeric, sections for, 949  
numeric, standard, 947–948

FormatFlags **property** (StringFormat **class**),  
**597–601, 642**

**formatting strings. See format specifiers**

**formatting text. See also text**

format flags for, 597–601  
positioning, 595–597  
rotating at any angle, 600–601  
tab stops, 601–602  
trimming, 602–605  
vertical text, 598–599

**forms. See also Windows forms controls**

About dialogs, 279–281  
adding components at design time, 68–70  
as controls, 275  
creating controls at design time, 70–73  
creating controls at run time, 73–75  
dialogs, 302–303  
event sequence for controls, 773  
icons for, 282–283  
login forms, 279–281  
MDI applications, 290–295  
MDI versus SDI applications, 295  
mouse cursors, 281–282  
MRU (Most Recently Used) lists, 296–302  
overriding WndProc routine, 285–289  
overview, 304  
properties adopted by child controls, 284  
property reset methods, 285  
SDI applications, 289, 292, 295  
sections, 108–109  
splash screens, 279–281  
transparency properties for, 275–279  
wizards, 303–304

#### Formula Workshop, 669–670

#### Friend keyword

for classes, 424  
for constants, 163, 164  
for events, 449  
for subroutines, 192–193  
for variables, 138

#### functions

for data type conversion (table), 135–136, 778–779  
declaring, 202–203, 787–788  
defined, 187

delegates, 165–166  
overloading, 420  
parameter declarations, 157–159  
scope of variables, 155  
setting return values, 202–203  
side effects, 208  
syntax, 202  
for validating data, 237–238

## G

#### garbage collection

brushes and, 590–591  
Dispose method, 444–446, 590–591  
Finalize method, 442–444  
overview, 442

**GDI+ (Graphics Device Interface+), 521, 523. See also drawing**

#### generics

advantages of, 511–512  
code example, 925  
collection classes, 507–508  
constrained types with, 516–518  
constructors, 514  
data type, 925  
declaring classes, 925  
defining, 512–514  
derived classes, 519  
Imports statement with, 519  
multiple types with, 514–516  
overview, 511, 520  
predefined classes, 519  
using, 518–519

GetAllSettings **method**, **682–683**

GetDataPresent **method**, **393, 395**

GetFormats **method**, **394**

GetPixel **method**, **625–626, 630**

GetSetting **method**, **682**

GetTabStops **method**, **601–602**

GoTo **statement**, **228–231, 793**

Graphics **class. See also drawing**

advanced transformations, 551–556  
anti-aliasing, 542–546  
DrawBezier method, 534–535  
DrawBeziers method, 535–536  
DrawClosedCurve method, 535–538  
drawing methods (table), 533–534, 931  
DrawPath method, 538–539  
DrawString method, 593–595  
filling methods (table), 931  
Graphics object, 532  
InterpolationMode property, 544–545  
MeasureCharacterRanges method, 606–609  
MeasureString method, 605–607, 642  
overloaded methods in, 523  
overview, 532–533  
properties and methods (table), 540–542, 932–933  
ResetTransform method, 647, 649  
saving and restoring graphics states, 556–558  
ScaleTransform method, 650

## Graphics class (continued)

SmoothingMode property, 543  
TextRenderingHint property, 543–544  
transformation basics, 546–551  
TranslateTransform method, 647  
**graphics classes. See drawing classes; specific classes**  
**Graphics Device Interface+ (GDI+), 521, 523. See also drawing**  
**GraphicsPath class or object**  
defining clipping regions using, 588  
defining form regions using, 586–588  
drawing and filling, 585–586  
properties and methods, 588–590, 935  
**GraphicsState object, 556**  
**Group Expert dialog, 668**  
**GroupBox control, 265, 826**  
**gutter for printing, 640–641**

## H

**“has-a” relationships, 414–415**  
**HashTable class, 496–497**  
**HatchBrush class, 572, 574–575**  
**HatchStyle enumeration, 527, 575–576**  
**heap**  
classes and, 434  
garbage collection, 442  
managed, 442  
performance and, 434  
**Help menu**  
How Do I command, 48  
overview, 47  
**HelpNavigator property values, 827**  
**HelpProvider component, 826–828**  
**hidden files**  
list of, 105  
modifying, 106  
purposes of, 104–106  
showing, 104  
**highlighting in code editor, 58–59**  
**How Do I command (Help menu), 48**  
**HScrollBar control**  
for initiating action, 271  
for making selections, 267  
overview, 828  
**HybridDictionary class, 497**

## I

**icons**  
application icons, 283  
in code editor margins, 52–53  
displaying in system tray, 283, 852–854  
editor for, 282–283  
for forms, 282–283  
for MDI container forms, 290  
setting for custom controls, 363  
**IDE (integrated development environment). See also specific pieces**  
avoiding customizing, 2, 3, 41

code editor (overview), 51–64  
complexity of, 1  
customizability of, 1–2  
key pieces, 4  
menus (overview), 4, 5  
overview, 3–5, 65  
projects, 2–3  
Properties Page (overview), 16–18  
rearranging secondary windows, 48–49  
solutions, 2–3  
toolbars, 4, 48  
Toolbox (overview), 4, 49–51, 802–804  
**If Then statements**  
I If statement versus, 212–214, 791  
multiline, 207–208, 790  
Select Case statement versus, 209–210  
single-line, 205–206, 789–790  
**I If statement, 212–214, 791**  
**Image class**  
overview, 615, 632  
properties and methods (table), 616–617, 936–937  
**image processing. See also drawing; specific classes**  
Bitmap class for, 617–630  
Image class for, 615–617  
Metafile class for, 630–632  
overview, 632–633  
**ImageList component**  
overview, 272, 829  
for TabControl control, 80–82  
**immed command, 32**  
**Immediate window**  
opening, 28, 32  
overview, 32–33  
**Implements keyword**  
in class declarations, 427–431  
in event declarations, 449  
**Imports statement**  
accessing namespaces inside files, 120, 462–463  
aliases with, 465–466  
automatic imports without, 464–465  
for generics, 519  
for namespace elements, 466  
**IndexOf method, 481–482**  
**Info object (My.Application)**  
properties (table), 959–960  
setting values, 960–962  
**Info object (My.Computer), 966–967**  
**information hiding. See encapsulation**  
**inheritance**  
adding and modifying class features and, 415–416  
child as refinement of parent, 412, 413  
for child controls on forms, 284  
in class declarations, 425–426  
code reuse and, 411–412  
from Component class, 801  
from Control class, 84–85, 801  
for custom controls, 380  
defined, 410  
deriving the child class, 410  
Form class trail for, 275  
“has-a” and “is-a” relationships, 414–415

hierarchies, 411–412  
 interface, 412, 417, 430–431  
 multiple, 412, 429–430  
 overview, 410–411  
 parent as abstraction of children, 412–414  
 subclassing the parent class, 410

**inheritance clause for classes, 425–426**  
**inheritance\_mode clause for subroutines, 181–182**  
**Inherits statement in class declarations, 427**  
**initialization\_expression clause for constants, 164**  
**initialization\_expression clause for variables**  
 initializing arrays, 148–150  
 initializing object arrays, 150  
 overview, 148

**instantiation**  
 of classes, 409, 438–440  
 of structures, 440–441

**integrated development environment. See IDE**

**IntelliSense**  
 in code editor, 56–57  
 Command window support for, 33  
 generic classes and, 511

**interfaces**  
 defined, 200  
 FileSystemNeeded property, 428  
 for groups with For Each statement, 220  
 implemented by classes, 427–430  
 implemented by structures, 431–432  
 implemented by subroutines, 200–201  
 inheritance, 412, 417, 430–431  
 naming conventions, 427  
 public interface of a class, 409–410

**Internet resources**  
 Business Objects site, 655  
 connection object information, 318–319  
 cryptography information, 751  
 Crystal Reports for Visual Studio 2005 information, 655  
 regular expression information, 743  
 third-party controls, 273  
 transaction object information, 325–326

**InterpolationMode property (Graphics class), 544–545**  
**InvisibleControl class, 379–380**  
**Is operator, 171**  
**“is-a” relationships, 414–415**  
**IsNot operator, 171**  
**Items.Add method, 809–810**  
**iterators, 224–225**

## J

**joining lines of code, 126–127**

## K

**Keyboard object (My.Computer), 967**  
**keyboard shortcuts, 43–44**

## L

**Label control, 268, 829**  
**labels for code lines, 127**  
**LastIndexOf method, 481–482**  
**libraries**  
 encapsulation by, 410  
 removing unused references, 465  
 Windows Control Library template, 362

**lifetime of variables, 129**  
**LIFO lists. See stacks (data structures)**  
**Like operator**  
 characters with special meaning, 171–172, 782  
 overview, 171–172  
 patterns, 172, 783

**line continuation**  
 comments and, 121  
 overview, 126

**line joining, 126–127**  
**line labels, 127**  
**LineAlignment property (StringFormat class), 595**  
**LinearGradientBrush class, 572, 575–580**  
**LineCap enumeration, 527**  
**LineJoin enumeration, 527**  
**linked lists, ListDictionary class for, 494–496**  
**LinkLabel control, 271, 830–831**  
**ListBox controls**  
 binding, 359  
 moving and copying items between, with drag and drop, 392–393  
 moving items between, with drag and drop, 389–392  
 overview, 267, 831–833  
 properties (table), 832

**ListDictionary class**  
 overview, 494–496  
 for strongly typed dictionaries, 500

**ListView control**  
 creating items and subitems, 834–835  
 defining columns, 836  
 displaying check boxes, 835–836  
 helper code for, 837–839  
 manipulating items at run time, 837–839  
 overview, 268, 833–837  
 properties (table), 837  
 RowSortingListView derived from, 370–374  
 Sorting property, 839–841  
 View property, 833–834

**literal type characters, 133, 778**  
**loading bitmaps, 618–619**  
**localization resources, 696–697**  
**Locals window, 28**  
**Location property, 77–78**  
**LockBits method, 627**  
**logical operators, 172–174, 783**  
**login forms, 279–281**  
**looping statements**  
 Continue statement, 228  
 defined, 216, 791  
 Do Loop, 225–227, 792

## looping statements (continued)

- enumerators, 223–224
- Exit statement, 227–228
- For Each, 220–225, 791
- For Next, 216–219, 791
- iterators, 224–225
- overview, 216
- While End, 227, 792–793

## M

### Macro Explorer command (Tools menu), 38

### Macro Explorer window, 38

#### macros

- adding to menus or toolbars, 41
- alternative approaches to problems, 40
- recording temporary, 38
- running temporary, 38
- saving temporary, 38
- uses for, 40

### Macros submenu (Tools menu)

- Macro Explorer command, 38
- overview, 38–40
- Record TemporaryMacro command, 38
- Run TemporaryMacro command, 38

### managed heap, 442

#### MaskedTextBox control

- events (table), 845
- Mask property characters, 842–843
- overview, 268, 842–845
- properties (table), 844

### MDI (multiple-document interface) applications

- building, 292–293
- child forms and, 290–291, 293
- defined, 290
- events, 293–295
- features, 290–293
- implementing commands, 291–292
- SDI applications versus, 295
- Window menu for child forms, 292, 293

### Me object, My namespace versus, 676

### MeasureCharacterRanges method, 606–609

### MeasureString method, 605–607, 642

#### memory

- garbage collection, 442–446
- heap, 434, 442
- stack, 434
- structures versus classes and, 433–434

### MemoryStream class, 710–711

#### menus. See also custom menus

- adding macros to, 41
- Build, 18–22
- creating new menus, 41
- Data, 33–34
- Debug, 22–33
- Edit, 8–10
- File, 5–8
- Format, 34–36
- Help, 48
- overview, 4, 5
- Project, 12–18

- removing commands, 41
- right-click menus, 42–43, 50–51
- Tools, 36–45
- View, 10–12
- Window, 45–47

### MenuStrip control, 270, 845–847

### MessageQueue component, 847

#### Metafile class

- building a Metafile, 630
- methods (table), 630, 938
- overview, 630, 633
- program demonstrating, 630–632

#### methods. See also methods, tables summarizing; specific methods

- assigning attributes for custom controls, 368
- of command objects, 331–332
- of controls, 90, 766–768
- declaring within classes, 446–447
- defined, 407
- delegates, 165–166
- events versus, 448
- implementing for custom controls, 366–367
- overloading, 418–420
- for Registry access, 682–683
- shared, 456–458
- for transformations (Graphics class), 547, 551

#### methods, tables summarizing

- Application class, 702, 955
- ArrayList class, 486–487
- Audio object (My.Computer), 963
- BinaryReader class, 712, 973–978
- BinaryWriter class, 713, 978
- Bitmap class, 617–618, 936–937
- Clipboard object (My.Computer), 404–405, 963–964
- Clock object (My.Computer), 964
- CollectionBase class, 489
- connection objects, 322
- Control class, 766–768
- CurrencyManager object, 353
- DataRow object, 340–341
- DataSet component, 335–336
- DataTable object, 338
- DataGridView object, 350
- dictionary methods, 494
- Directory class, 723–724, 985–986
- DirectoryInfo class, 727–729, 989–990
- Exception class, 242
- File class, 724–726, 986–988
- file methods, 718, 981–982
- FileInfo class, 729–730, 991–992
- FileSystem class (System.IO), 983–985
- file-system methods, 719–720, 982–983
- FileSystem object (My.Computer), 735–736, 965–966, 992–997
- FileSystemWatcher class, 992
- Form class, 911–914
- Graphics class, 533–534, 540–542, 931–933
- GraphicsPath class, 589, 935
- Image class, 616–617, 936–937
- LinearGradientBrush class, 575–577
- Metafile class, 630, 938
- MonthCalendar control, 850–851

- My.Application, 957–962
  - My.Settings namespace, 688
  - My.User namespace, 974
  - NameValueCollection class, 492
  - Network object (My.Computer), 968
  - OleDbConnection class, 322
  - Path class, 733–734, 990–995
  - PathGradientBrush class, 580–582
  - Pen class, 564–565, 933–934
  - queue methods, 505
  - Registry class (My.Computer), 684, 971–972
  - RegistryKey class, 684, 971–972
  - RichTextBox control, 875–876
  - Screen object (My.Computer), 972–973
  - Screen object (My.User), 974
  - SerialPort class, 969–970
  - SqlConnection class, 322
  - stack methods, 503
  - Stream class, 709, 976–977
  - StringFormat class, 596–597, 936
  - strongly typed dictionaries, 498
  - System.Environment object, 680–681
  - TableLayoutPanel control, 888
  - TextBox control, 890–891
  - TextReader class, 713, 979
  - TextureBrush class, 572–573
  - TextWriter class, 714, 979
  - ToolTip component, 895
  - TreeNode objects, 901–902
  - TreeView control, 899
  - WebBrowser control, 902
  - Microsoft**
    - Visual Studio Tools for Office (VSTO), 636
    - Word, printing from, 636
  - Microsoft namespace, 740**
  - Module **statement**
    - code regions defined by, 109
    - top-level nodes defined by, 108
  - modules**
    - code regions, 109–110
    - namespaces versus, 469–470
    - scope of variables, 155–156
    - sections, 108–109
    - simple example, 108
  - Modules window, 29–30**
  - MonthCalendar **control**
    - clearing current day's date, 849
    - methods (table), 850–851
    - overview, 267, 847–852
    - properties (table), 848–849
  - Most Recently Used lists. See MRU lists**
  - mouse cursors, 281–282**
  - mouse events. See also drag and drop**
    - drag source events, 387
    - drop target events, 386–387
    - sequence, 772
  - Mouse **object (My.Computer), 967**
  - move events, sequence for, 773**
  - MRU (Most Recently Used) lists**
    - building, 296–300
    - defined, 296
    - main program use of, 300–302
  - multiline If Then statement**
    - function side effects and, 208
    - overview, 207–208, 790
  - multiple inheritance, 412, 429–430**
  - multiple-document interface applications. See MDI applications**
  - multithreading, BackgroundWorker component for, 805**
  - My namespace**
    - Me object versus, 676
    - My.Application, 957–962
    - My.Computer, 962–973
    - My.Computer.FileSystem object, 734–736, 965–966, 995–998
    - My.Forms, 973–974
    - My.Resources, 974
    - My.User, 970
    - need for, 675–676
    - sections, 677
  - My.Application namespace**
    - Application class versus, 700
    - Info object properties (table), 959–960
    - overview, 957
    - properties, methods, and events (table), 957–962
    - setting Info values, 960–962
  - My.Application.AssemblyInfo namespace, 107**
  - My.Computer namespace**
    - Audio object, 963
    - Clipboard object, 963–964
    - Clock object, 964
    - FileSystem object, 734–736, 965–966, 992–997
    - Info object, 966–967
    - Keyboard object, 967
    - Mouse object, 967
    - Name property, 968
    - Network object, 968
    - overview, 962
    - Ports object, 968–970
    - Registry object, 970–972
    - Screen object, 972
  - My.Computer.FileSystem **object**
    - methods (table), 735–736, 965–966, 992–997
    - properties (table), 734–735, 965–966, 995
    - SpecialDirectoriesProxy object, 737, 997–998
  - My.Computer.FileSystem.  
SpecialDirectoriesProxy **object, 737, 997–998**
  - My.Computer.Registry namespace**
    - overview, 683
    - properties (table), 683
    - RegistryKey class, 683–685, 971–972
  - My.Forms namespace, 973–970**
  - My.Resources namespace, 970**
  - My.Settings namespace, 688–689**
  - My.User namespace, 970**
- ## N
- name **clause for variables, 145–146**
  - name collisions, 459, 461**
  - Name **property (My.Computer), 968**

# Namespace statement

---

## Namespace statement, 467–469

### namespaces. *See also specific namespaces*

- aliases, 465–466, 519
- automatic imports, 464–465
- classes, structures, and modules versus, 469–470
- creating, 467–469
- defined, 118
- defining, 118–119
- Direct3D, 756–762
- drawing namespaces, 523–532, 927–930
- high-level .NET Framework namespaces, 740–742
- importing elements, 466
- Imports statement for, 120, 462–467
- Microsoft namespace, 740
- name collisions avoided by, 459, 461
- overview, 473
- for projects, 119
- referencing modules, 119
- reflection, 751–755
- regular expressions, 742–745
- resolving, 470–473
- root namespace, 467
- scope of variables, 156
- System namespace, 740–742
- System.Security, 747–751
- System.Xml, 746–747

### NameValueCollection class, 491–493

### naming conventions for variables, 166–167

### .NET Framework

- Directory class, 723–724, 985–986
- DirectoryInfo class, 727–729, 989–990
- Direct3D, 756–762
- DriveInfo class, 726–727, 988–989
- File class, 724–726, 986–988
- FileInfo class, 729–730, 991–992
- FileSystemInfo class, 731
- FileSystemWatcher class, 731–733, 992
- high-level namespaces, 740–742
- Microsoft namespace, 740
- overview, 723, 739, 762
- Path class, 733–734, 990–995
- reflection, 751–755
- regular expressions, 742–745
- System namespace, 740–742
- System.Security namespace, 747–751
- System.Xml namespace, 745–747

### Network object (My.Computer), 968

### New File dialog, 6, 7

### New keyword

- for class instantiation, 438
- for variables, 147–148

### New Solution Configuration dialog, 21

### nonclient area of controls, 89

### notification icons, 283, 852–854

### NotifyIcon component, 269, 283, 852–854

### numeric format specifiers

- custom, 948–949
- sections, 949
- standard, 947–948

### NumericUpDown control, 267, 854–855

## O

### Object class, 409

### object-oriented programming concepts. *See OOP concepts*

- objects
  - class instantiation, 409, 438–440
  - defined, 409
  - Dispose method, 444–446
  - Finalize method, 442–444
  - polymorphism, 417–418
  - structure instantiation, 440–441

### Of type\_list clause

- for classes, 426
- for generics, 925

### OleDbConnection class, 319, 322

### On Error statements

- exiting error-handling code, 799
- On Error GoTo line, 250–251, 799
- On Error GoTo -1, 252–253, 799
- On Error GoTo 0, 252, 799
- On Error Resume Next, 251–252, 799

### OOP (object-oriented programming) concepts

- abstraction, 412–414
- classes, 407–409
- encapsulation, 409–410
- inheritance, 410–417
- overloading, 418–420
- overview, 420
- polymorphism, 417–418
- refinement, 412, 413

### Opacity property (Form object), 275–277, 279

### OpenFileDialog component, 855–857

### operators

- arithmetic, 169–170, 781
- assignment, 175–176, 784
- bitwise, 174, 783
- comparison, 170–172, 782–783
- concatenation, 170, 782
- Date data type operations, 178–181, 785
- defined, 169
- logical, 172–174, 783
- overloading, 181–184, 786
- overview, 184–185
- precedence of, 174–175, 784
- StringBuilder class for concatenation, 176–178
- TimeSpan data type operations, 179–180, 785

### Option Compare compiler option, 18

### Option Explicit compiler option

- debugging and, 18
- performance and, 152–153
- problems from keeping turned of, 152–153
- variable declarations and, 16, 151–153, 776

### Option Strict compiler option

- constant declarations and, 164
- data type conversion and, 17
- debugging and, 18
- variable declarations and, 153, 776

### Optional keyword for subroutines, 197–198

### Options dialog, 43–44

### order of precedence for operators, 174–175, 784

**outlines**, **DrawPath** method for, 538–539

**outlining in code editor**, 53–55

**overloading**

functions, 420

methods, 418–420

operators, 181–184, 786

overloaded **Graphics** methods, 523

overloaded stack constructors, 503–504

subroutines, 191, 198

**overriding**

subroutines, 191–192

**WndProc** routine, 285–289

## P

**PageSetupDialog** component, 858–859

**Paint** event handler

drawing events and, 558–560

making forms do all drawing in, 561

overview, 522

**PaintEventArgs** parameter, 558–559

transformations and, 548, 549, 554

**Panel** control, 265–266, 859–860

**parameter declarations**

**ByRef**, 157, 158–159, 194–196

**ByVal**, 157–158, 193–194, 195–196

declaring arrays **ByVal** and **ByRef**, 195–196

**Optional** keyword, 197–198

overloading and optional parameters, 198

overview, 157–159

parameter arrays for subroutines, 198–200

parenthesized parameters, 196–197

for subroutines, 193–200

**parameter passing**, 436–437

**parameters clause for events**, 449

**parenthesized parameters for subroutines**, 196–197

**parsing methods for data types**, 137, 779

**Partial** keyword

in class declarations, 422–423

code regions and, 110

**Path** class, 733–734, 990–995

**PathGradientBrush** class, 572, 580–585

**patterns for Like** operator, 172, 783

**Pen** class

**Alignment** property, 566–567

**Brush** class versus, 563

**CompoundArray** property, 567–568

constructors (table), 564

**CustomEndCap** property, 567–568

**CustomStartCap** property, 567–568

drawing lines one pixel wide, 563–564

overview, 521, 563

properties and methods (table), 564–565, 933–934

transformations, 569–571

**percent sign (%) preceding custom format specifiers**, 946

**performance**

anti-aliasing and, 546

**Choose** versus **Select Case** and, 215

connection objects and, 319

data types and, 131

heap versus stack and, 434

**If Then Else** versus **Select Case** and, 209–210

normal arrays versus **Array** objects, 479–481

**Option Explicit** compiler option and, 152–153

pixel manipulation and, 630

structures versus classes and, 434

**PerformanceCounter** component, 860

**PictureBox** control

creating drawings, 861

keeping drawings visible, 861–862

overview, 271, 860–862

**SizeMode** property values (table), 860–861

**pixel-by-pixel operations**

**BitmapBytesRGB24** class for, 627–630

**GetPixel** method for, 625–626, 630

**SetPixel** method for, 625–626, 630

“unsafe” pixel manipulation, 627–630

**pointer (Toolbox)**, 804

**polymorphism**, 417–418

**popping items onto and off of stacks**, 502–503

**Ports** object (**My.Computer**)

property and method, 968

**SerialPort** class, 968–970

**position properties of controls**, 89–90

**precedence of operators**, 174–175, 784

**Print** method, 636, 637, 654

**PrintDialog** component

overview, 862–865

**PrintDocument** component and, 638, 865

**PrinterSettings** object properties for (table), 864–865

properties (table), 863

**PrintDocument** component

basic process, 635

**BeginPrint** event handler, 641

calling **Print** method, 636

**EndPrint** event handler, 643

events, 636–637

overview, 271, 272, 866

**Print** method, 637, 654

**PrintDialog** component and, 638, 865

printing shapes with, 636–640

printing text with, 641–646

**PrintPage** event handler, 642, 647, 652

**PrintPreviewControl** control and, 637

**PrintPreviewDialog** component and, 633, 638

**QueryPageSettings** event handler, 642

simple printing application, 638–640

**PrinterSettings** object

**PrintDialog** component and, 863

properties (table), 864–865

**printing**. **See also** **PrintDocument** component

basic process, 635, 636–640

centering printouts, 647–649

complexity of, 635–636

fitting pictures to the page, 649–651

gutter for, 640–641

overview, 635, 654

**PageSetupDialog** component for, 858–859

**PrintDialog** component for, 638, 862–865

**PrintPreviewControl** control for, 271, 633, 866–868

## printing (continued)

PrintPreviewDialog component for, 271, 633, 638, 868–869  
simple printing application, 638–640  
simplifying drawing and, 651–654  
text, 640–646  
tools for, 804  
with Visual Studio Tools for Office, 636  
with word processing and text-formatting applications, 636

## PrintPreviewControl control

overview, 271, 866–868  
PrintDocument component and, 633  
properties (table), 868

## PrintPreviewDialog component

overview, 271, 868–869  
PrintDocument component and, 633, 638

## Private keyword

for classes, 424  
for constants, 163, 164  
for events, 449  
for subroutines, 193  
for variables, 129, 138, 150–151

## procedures. See also functions; subroutines

defined, 187  
property procedures, 157–159, 203–204, 788  
scope of variables, 155

## Process component, 869–870

## Professional Crystal Reports for Visual Studio .NET (Wiley publication), 655

## program control statements. See also specific kinds

decision statements, 205–215, 789–791  
defined, 205, 789  
GoTo, 228–231, 793  
looping statements, 216–228, 791–793

## ProgressBar control, 270, 870–871

## Project menu, 12–15

## projects. See also code file structure

for creating custom controls, 362  
custom controls in executable projects, 380–382  
defined, 2, 103  
excluding from builds, 22  
hidden files, 103–107  
overview, 2–3  
referring to variables or routines, 108  
for testing custom controls, 364–365

## properties. See also properties, tables summarizing

for anti-aliasing, 543–545  
assigning attributes for custom controls, 368–369  
binding (complex strategy), 356–359  
binding (simple strategy), 351–352  
of child controls, 284  
collection properties, 80–83, 84  
of command objects, 330–331  
compound properties, 77–78  
declaring within classes, 446–447  
defined, 75, 407  
events changed by (Form class), 918–919  
events versus, 448  
implementing for custom controls, 366–367  
inheritance from Control class, 84–85  
My.Computer.Registry namespace, 683  
position and size properties, 89–90

read-only properties, 84  
resetting in forms, 285  
restricted properties, 78–80  
setting at design time, 76–83  
setting at run time, 83–84  
specialized editors for, 78, 79–80

## Properties Page

Compile tab, 16–18  
displaying, 11–12, 15  
Option Compare, 18  
Option Explicit, 16, 18, 776  
Option Strict, 17–18, 776  
overview, 16, 17

## properties, tables summarizing

Application class, 701, 953–954  
ArrayList class, 486–487  
Clock object (My.Computer), 964  
connection objects, 319  
control and component properties, 763–766  
Control class, 763–766  
CurrencyManager object, 352–353  
DataColumn object, 341–342  
DataRelation object, 344  
DataRow object, 339  
DataRowView object, 350–351  
DataSet component, 334  
DataTable object, 337  
DataView object, 349–350  
dictionary properties, 494  
DirectoryInfo class, 727–729, 989–990  
DriveInfo class, 727, 988–989  
Err object, 256  
Exception class, 242  
FileInfo class, 729–730, 991–992  
FileSystem class (System.IO), 983–985  
FileSystem object (My.Computer), 734–735, 965–966, 995  
FileSystemWatcher class, 731, 992  
FontDialog component, 824–825  
ForeignKeyConstraint, 346  
Form class, 905–911  
Graphics class, 540–542, 932–933  
GraphicsPath class, 589, 935  
Image class, 616–617, 936–937  
Info object (My.Application), 959–960  
Info object (My.Computer), 966–967  
Keyboard object (My.Computer), 967  
LinearGradientBrush class, 575–577  
LinkLabel control, 830  
ListBox control, 832  
ListView control, 837  
MaskedTextBox control, 844  
MenuStrip control, 846  
MonthCalendar control, 848–849  
Mouse object (My.Computer), 967  
My.Application, 957–962  
My.Computer.FileSystem object, 734–735, 965–966, 995  
My.Computer.FileSystem.SpecialDirectoriesProxy object, 737, 997–998  
My.Computer.Registry namespace, 683

- My.Settings namespace, 688
  - My.User namespace, 970
  - NameValueCollection class, 492
  - NumericUpDown control, 854
  - OleDbConnection class, 319
  - OpenFileDialog component, 856–857
  - PageSetupDialog component, 859
  - Path class, 733, 994
  - PathGradientBrush class, 580–582
  - Pen class, 564–565, 933–934
  - PrintDialog component, 863
  - PrinterSettings object, 864–865
  - PrintPreviewControl control, 868
  - queue properties, 505
  - RadioButton control, 872
  - Registry class (My.Computer), 684, 971–972
  - RegistryKey class, 684, 971–972
  - RichTextBox control, 873–875
  - Screen object (My.Computer), 972
  - Screen object (My.User), 970
  - SerialPort class, 968–969
  - SplitContainer control, 880
  - SqlConnection class, 319
  - stack properties, 503
  - Stream class, 708, 976
  - StringFormat class, 596–597, 936
  - System.Environment object, 679–680
  - TabControl control, 883–885
  - TableLayoutPanel control, 887–888
  - TextBox control, 889–890
  - TextureBrush class, 572–573
  - TextWriter class, 979
  - ToolStrip control, 893
  - ToolTip component, 894–895
  - TrackBar control, 896
  - TreeNode objects, 900–901
  - TreeView control, 898–899
  - UniqueConstraint, 347
  - WebBrowser control, 902
  - Properties window**
    - compound properties and, 77–78
    - overview, 4
    - restricted properties and, 78–80
    - setting control properties, 76
  - Property Pages command (View menu), 11–12**
  - property procedures**
    - declaring, 160, 203, 788
    - defined, 159, 203
    - overview, 203–204
    - parameter declarations, 157–159
    - read-only, 160, 788
    - read/write, 788
    - uses for, 159–160
    - write-only, 160, 788
  - PropertyGet **procedures, 788**
  - PropertyGrid **control, 268–269, 871**
  - Protected Friend **keyword**
    - for classes, 424
    - for constants, 163, 164
    - for events, 448
    - for subroutines, 193
    - for variables, 138
  - Protected **keyword**
    - for classes, 423
    - for constants, 163
    - for events, 448
    - for subroutines, 192
    - for variables, 138
  - Public **keyword**
    - for classes, 423
    - for constants, 163, 164
    - for events, 448
    - for subroutines, 192
    - for variables, 129, 138, 150
  - Publish Wizard, 20**
- ## Q
- queues**
    - circular arrays in, 505
    - defined, 502
    - FIFO behavior, 502
    - properties and methods (table), 505
    - using, 506–507
  - QuickWatch dialog, 25**
- ## R
- RadioButton **control, 267, 871–872**
  - RaiseEvent **keyword, 450**
  - RaiseEvent **routine, 452–453**
  - raising events**
    - custom events, 452–453
    - defined, 90
    - RaiseEvent keyword for, 450
    - syntax, 923
  - random-file access, 720–722**
  - ReadOnly **keyword**
    - constants and, 164
    - for variables, 142–143, 150
  - read-only property procedures, 788**
  - read-only strongly typed collections, 491**
  - ReadOnlyCollectionBase **class, 491**
  - read/write property procedures, 788**
  - Record TemporaryMacro command (Tools menu), 38**
  - refinement, 412, 413**
  - reflection**
    - defined, 751
    - for learning about other applications, 753–755
    - listing program's properties, types, and values, 751–753
    - updates, 755
  - Regex **object, 743–745**
  - Region **statement, organizing code using, 109–110**
  - Registry**
    - accessing, 682
    - defined, 681
    - My.Computer.Registry namespace for, 683–685
    - subtrees (table), 681
    - uninstalling programs and, 683
    - unsupported keys, 681
    - Visual Basic methods for, 682–683, 685
  - Registry class (My.Computer)**
    - properties and methods (table), 684, 971–972
    - working with Registry using, 683, 684–685

Registry **object (My.Computer)**, 970–971  
RegistryKey **class**, 971–972  
regression testing tools, 234  
regular expressions, 742–745  
Relation editor, 312  
Release configurations, Debug configurations versus, 21  
RemoveHandler **statement**, 93–94, 452–453  
ReportDocument **object**, 672–673  
reporting  
  building a report, 656–666  
  Crystal Reports for Visual Studio 2005 for, 655–666  
  CrystalReportViewer control for, 666–667, 671–673  
  customizing reports, 667–671  
  manipulating output programmatically, 672–673  
  overview, 655, 673  
  ReportDocument object for, 672–673  
  .rpt extension for reports, 655  
  using external reports, 671–672  
  viewing reports, 656, 666–667  
ReportSource **property (CrystalReport Viewer control)**, 671  
ReportViewer **control**, 872  
ResetTransform **method**, 647, 649  
Resize **event handler**, 560–561  
resize events, **sequence for**, 772–773  
resolving namespaces, 470–473  
ResourceManager **class**, 694, 698  
resources. *See also* configuring applications  
  application settings versus, 104  
  ComponentResourceManager class for, 698–700  
  creating application resources, 689–692  
  defined, 104  
  embedded, 694–695  
  hidden files for, 104, 105  
  localization, 696–697  
  resource files overview, 689  
  ResourceManager class and, 694, 698  
  satellite, 695–696  
  strongly typed embedded, 695  
  unmanaged, handling, 442–446  
  using application resources, 692–694  
Resources.Designer.vb **file**, 692, 693  
Resources.resx **file**, 104, 105, 688  
restricted properties of controls, 78–80  
restricting scope of variables, 156–157  
return values  
  functions' setting of, 202–203  
  needed for PropertyGet procedures, 788  
RichTextBox **control**  
  methods (table), 875–876  
  overview, 267–268, 873–877  
  properties (table), 873–875  
right-click menus  
  Customize dialog, 42–43  
  Toolbox, 50–51  
root namespace, 467  
rotating text, 600–601  
routines. *See* functions; subroutines  
RowSortingListView **derived control**  
  building, 370  
  hiding parent features, 373–374

  implementing custom sort orders, 370–371  
  shadowing parent features, 372–373  
  .rpt **extension for reports**, 655  
Run TemporaryMacro **command (Tools menu)**, 38  
run time  
  code editor at, 54  
  control properties at, 83–84  
  controls versus components at, 67, 68  
  creating controls at, 73–75  
  managing custom controls for, 369  
  manipulating ListView control items, 837–839  
  methods invoked at, 90  
  retrieving environment variables, 679  
  setting event handlers at, 93–94  
RunWorkerAsync **method**, 805

## S

satellite resources, 695–696  
SaveFileDialog **component**, 877–878  
SaveSetting **method**, 682  
saving  
  bitmaps, 619–620  
  connection string, 308–310  
  graphics states, restoring and, 556–558  
  temporary macros, 38  
ScaleTransform **method**, 650  
Schema Editor, 311–312  
scope of classes (**unit scope**), 446–447  
scope of variables  
  block scope, 154–155  
  defined, 129, 154  
  module scope, 155–156  
  namespace scope, 156  
  procedure scope, 155  
  restricting, 156–157  
Screen **object**  
  My.Computer, 972–973  
  My.User, 970  
scroll bars  
  in ColorScroller composite control, 374–375  
  HScrollBar control for, 267, 828  
  for Panel control, 860  
  VScrollBar control for, 267, 902  
SDI (**single-document interface**) applications, 289, 292, 295  
Select Case **statement**  
  Choose statement versus, 214–215  
  comma-separated expressions, 211–212  
  If Then Else statement versus, 209–210  
  Is keyword, 210–211, 790  
  To keyword, 210, 790  
  overview, 208–210, 790  
  syntax, 208–209  
selecting  
  controls, deciding which to use, 264–272  
  controls from Toolbox, 72–73  
  making selections, controls for, 266–267  
separated validation, 97–98  
sequences of events, 772–773  
sequential-file access, 720

- SerialPort class**  
 events (table), 970  
 methods (table), 969–970  
 properties (table), 968–969
- SerialPort component, 878**
- ServiceController component, 879**
- SetPixel method, 625–626, 630**
- SetTabStops method, 601–602**
- Settings.settings file, 104, 105**
- Shadows keyword**  
 for classes, 424–425  
 for derived controls, 372–374  
 for events, 449  
 for subroutines, 192  
 for variables, 139–140, 150
- Shared keyword**  
 constants and, 164  
 for methods, 456–458  
 for subroutines, 192  
 for variables, 139–140, 150, 455–456
- ShowUpDown property (DateTimePicker control), 814–815**
- side effects of functions, 208**
- SimpleSmiley custom control, 376–377**
- single-document interface (SDI) applications, 289, 292, 295**
- single-line If Then statement**  
 Else clause, 206, 789, 790  
 Else If clause, 789  
 overview, 205–206, 789–790
- size properties of controls, 89–90**
- SizeMode property values (PictureBox control), 860–861**
- SmoothingMode property (Graphics class), 543**
- snippets. See code snippets**
- SolidBrush class, 572**
- Solution Explorer**  
 database application in, 311, 312  
 external reports in, 671  
 Imported Namespaces list, 464–465  
 overview, 4  
 Unused References list, 465
- Solution Properties Page dialog, 18, 19**
- solutions, 2–3**
- SortedList class, 500–501**
- Sorting property (ListView control), 839–841**
- special characters. See characters; operators**
- speed. See performance**
- splash screens, 279–281**
- SplitContainer control**  
 events, 881  
 overview, 266, 879–881  
 properties (table), 880
- SqlConnection class, 319, 322**
- stack (memory), 434**
- stacks (data structures)**  
 defined, 502  
 LIFO behavior, 502  
 memory management, 503–504  
 overloaded constructors, 503–504  
 popping items onto and off of, 502–503  
 properties and methods (table), 503
- StackTrace objects, 243–246**
- Standard Report Creation Wizard**  
 Chart page, 664  
 connection wizard, 659–660  
 creating a new data source, 657–659  
 Fields page, 660–661  
 Group Sorting page, 662–663  
 Grouping page, 662  
 Link page, 660–661  
 opening, 657  
 Record Selection page, 664, 665  
 Report Style page, 664–665  
 Summaries page, 662, 663
- statements. See also specific statements**  
 control statements, 789–793  
 decision statements, 205–215, 789–791  
 looping statements, 216–228, 791–793  
 in subroutine declarations, 201–202
- Static keyword**  
 constants and, 164  
 for variables, 138
- StatusStrip control, 270, 881–882**
- Stephens, Rod (*Visual Basic Graphics Programming*), 762**
- Stream class, 708–709, 976–977**
- StreamReader class, 715–716, 980**
- streams. See also specific classes**  
 classes (table), 708, 975–976  
 console streams, 980  
 custom stream classes, 716  
 defined, 707  
 overview, 707
- StreamWriter class, 715–716, 980**
- string formats. See format specifiers**
- StringBuilder class, concatenation using, 176–178**
- StringCollection class, 488**
- StringDictionary class, 500**
- StringFormat class**  
 Alignment property, 595, 642  
 FormatFlags property, 597–601, 642  
 GetTabStops method, 601–602  
 LineAlignment property, 595  
 positioning text using, 595–597  
 printing and, 642  
 properties and methods (table), 596–597, 936  
 SetTabStops method, 601–602  
 Trimming property, 602–605
- StringReader class, 714–715, 980**
- StringWriter class, 714–715, 980**
- strongly typed classes**  
 collections, 488–491  
 dictionaries, 497–500  
 generic classes as, 511  
 read-only strongly typed collections, 491
- strongly typed embedded resources, 695**
- structured error handling**  
 classic error handling versus, 254–256  
 custom exceptions for, 248–249  
 Exception objects for, 241–243  
 StackTrace objects for, 243–246  
 throwing exceptions, 246–248  
 Try block for, 239–241, 795

## structures

- boxing and unboxing by, 437
- classes versus, 431, 432–437, 458, 922
- constructors, 440–441, 922
- declaring, 431, 922
- inheritance not possible for, 431
- instantiation, 440–441
- interface implementation, 431–432
- memory requirements, 433–434
- name collisions and, 459
- namespaces versus, 469–470
- object assignment by, 435–436
- parameter passing by, 436–437
- performance and, 434
- stack and, 434
- syntax, 431, 922
  - in System.Drawing namespace (table), 525, 927–928
  - as value types, 432–433

**subroutines. See also constructors**

- accessibility clause, 192–193
- attribute\_list clause, 188–190
- declaring, 187–202, 787
- declaring arrays ByVal and ByRef, 195–196
- defined, 187
- delegates, 165–166
- executing in Immediate window, 33
- implementing interfaces, 200–201
- inheritance\_mode clause, 181–182
- optional parameters, 197–198
- overloading, 191, 198
- overriding, 191–192
- parameter arrays, 198–200
- parameter declarations, 157–159, 193–200
- parenthesized parameters, 196–197
- scope of variables, 155
- statements section, 201–202
- subroutine\_name clause, 193
- syntax, 187

**System namespace, 740–742**

**System Registry. See Registry**

**system tray, displaying icons in, 852–854**

**System.Drawing namespace**

- classes (table), 524, 927–928
- overview, 523, 525
- structures (table), 525, 927–928

**System.Drawing.Drawing2D namespace, 525–526, 928–929**

**System.Drawing.Imaging namespace, 528, 929**

**System.Drawing.Printing namespace, 532, 930**

**System.Drawing.Text namespace**

- classes (table), 529, 930
- displaying text with and without anti-aliasing, 531
- enumerations (table), 530
- listing system fonts, 529–530
- overview, 529

**System.Environment object, 679–681**

**System.IO namespace**

- Directory class, 723–724, 985–986
- DirectoryInfo class, 727–729, 989–990
- DriveInfo class, 726–727, 988–989
- File class, 724–726, 986–988

- FileInfo class, 729–730, 991–992
- FileSystem class, 983–985
- FileSystemInfo class, 731
- FileSystemWatcher class, 731–733, 992–993
- Path class, 733–734, 994–995

**System.Security namespace, 747–751**  
**System.Xml namespace, 746–747**

## T

### tab stops, 601–602

#### TabControl control

- ImageList component for, 80–82
- overview, 266, 882–886
- properties (tables), 883–885

#### TableLayoutPanel control

- FlowLayoutPanel control versus, 821, 889
- methods (table), 888
- overview, 265, 886–889
- properties (table), 887–888

#### TARGET constant, 113

#### testing custom controls

- test project for, 364–365
- UserControl Test Container for, 363–364
- using from Toolbox, 365–366

#### text

- determining where parts will be drawn, 606–609
- displaying with and without anti-aliasing, 531
- drawing, 593–595
- font metrics, 609–612
- formatting, 595–605
- listing system fonts, 529–530
- measuring width and height for fonts, 605–607
- overview, 593, 613
- positioning, 595–597
- printing, 640–646
- rotating at any angle, 600–601
- tab stops, 601–602
- trimming, 602–605
- vertical, 598–599

#### TextBox control

- methods (table), 890–891
- overview, 267–268, 889–891
- properties (table), 889–890

#### TextReader class, 713, 979

**TextRenderingHint property (Graphics class), 543–544**

#### TextureBrush class, 572, 572–575

#### TextWriter class, 713–714, 979

#### third-party controls, 272–273

#### Threads window, 29

#### 3D Computer Graphics (Watt, Allan H.), 762

#### Throw statement, 795

#### throwing events. See raising events

#### throwing exceptions, 246–248, 795

#### Timer component, 271, 891

#### times

- custom format specifiers, 944–946
- DateTimePicker control for, 814–816
- standard format specifiers, 943–944, 945, 946

#### TimeSpan data type operations, 179–180, 785

**toolbars. See also custom toolbars**

- adding macros to, 41
- overview, 4, 48
- rearranging, 48
- removing commands, 41
- showing or hiding, 48

**Toolbox. See also components; controls; specific tools**

- components listed by purpose, 803–804
- double-clicking on controls, 70
- dragging controls from, 70–71
- overview, 4, 49–51
- pointer, 804
- right-click menu for, 50–51
- setting icons for custom controls, 363
- standard components (table), 802–803
- using custom controls, 365–366
- Windows forms controls in, 259–264

**Tools menu**

- commands, 37–38
- Customize command, 40
- Macros submenu, 38–40
- overview, 36

**Tools.Alias command, 33****ToolStrip control, 270–271, 891–893****ToolStripContainer control, 893–894****ToolTip component, 269, 894–895****tooltips in code editor, 55–56, 58****Tostring method, 943–946****TRACE constant, 113, 116****TrackBar control**

- for initiating action, 271
- for making selections, 267
- overview, 896
- properties (table), 896

**transaction objects**

- further information, 325–326
- program demonstrating, 323–325
- transaction defined, 322

**transformations (graphical), 546**

- advanced, 551–556
- basics, 546–551
- combining, 547, 551
- defined, 546
- Graphics class methods for, 547, 551
- mapping world coordinates to device coordinates, 553
- matrices for, 546–547
- order of applying, 549, 551
- Pen class transformations, 569–571
- for positioning drawings, 554–556
- Prepend parameter for, 547–548
- rotation, 548–549
- rotation with translation, 549–551
- saving and restoring graphics states, 556–558
- scaling diamond around its center, 551–553

**TranslateTransform method, 647****transparency properties for forms****Opacity, 275–277, 279****TransparencyKey, 277–279****TransparencyKey property (Form object), 277–279****TreeNode objects**

- methods (table), 901–902
- overview, 897
- properties (table), 900–901

**TreeView control**

- methods (table), 899
- overview, 896–902
- properties (table), 898–899
- TreeNode object methods (table), 901–902
- TreeNode object properties (table), 900–901
- TreeNode objects, 897

**Trimming property (StringFormat class), 602–605****Try block**

- Catch statement, 239, 240–241
- for structured error handling, 239–241, 795
- syntax, 239, 795

**typeof operator, 171****types. See data types****typographic code elements**

- line continuation, 126
- line joining, 126–127
- line labels, 127
- normal comments, 121
- overview, 120–121
- XML comments, 122–125

**U****UniqueConstraint, 345, 346–347****unit scope of classes, 446–447****unmanaged resources**

- Dispose method for, 444–446
- Finalize method for, 442–444

**unplanned conditions. See also error handling**

- bugs versus, 233–234
- catching, 235–238
- defined, 234
- Y2K problem, 233

**UserControl class**

- for custom controls, 374, 380
- customizing in executable projects, 381
- inherited in executable projects, 381

**UserControl Test Container, 363–364****V****validating data. See also validation events**

- Assert method for, 235
- compiler constants for, 235
- functions for, 237–238
- need for, 234
- regression testing tools and, 234
- regular expressions for, 743–744
- testing for unexpected conditions, 236–237

**validation events**

- CausesValidation property for, 95, 97
- deferred validation, 98–100
- overview, 95
- separated validation, 97–98
- Validated events, 95, 97

## validation events (continued)

---

### validation events (continued)

- ValidateFiveDigits examples, 96–97, 98–100
- Validating event handlers, 95–100
- Validating events, 95, 97
- variable declarations**
  - accessibility clause, 138–139, 150
  - for arrays, using parentheses, 476
  - attribute\_list clause, 138
  - bounds\_list clause, 146–147
  - Dim keyword, 143
  - for enumerated types, 161, 776
  - initialization\_expression clause, 148–150
  - multiple declarations in single statement, 150–151
  - name clause, 145–146
  - naming conventions, 166–167
  - New keyword, 147–148
  - Option Explicit and, 16, 151–153, 776
  - Option Strict and, 153, 776
  - overview, 137–138, 168
  - parameter declarations, 157–159
  - ReadOnly keyword, 142–143
  - Shadows keyword, 140–142
  - Shared keyword, 139–140, 150, 455–456
  - standard declaration, 775–776
  - syntax, 137
  - WithEvents keyword, 74, 92–93, 143–145
- variables. See also data types; variable declarations**
  - accessibility, 129, 138–139, 150–151
  - debugging and, 18
  - delegates, 165–166
  - environment variables, 677–681
  - factors determining behavior of, 129–130
  - lifetime, 129
  - naming conventions, 166–167
  - Option Strict and conversion of, 17
  - read-only, 160
  - scope, 129, 154–157
  - write-only, 160
- VBC\_VER constant, 113
- vertical text, 598–599**
- View menu, 10–12**
- View property (ListView control), 833–834
- Visible property of controls, 379–380
- Visual Basic Graphics Programming* (Stephens, Rod), 762
- Visual Studio Tools for Office (VSTO) by Microsoft, 636**
- VScrollBar control
  - for initiating action, 271
  - for making selections, 267
  - overview, 902
- VSTO (Visual Studio Tools for Office) by Microsoft, 636**

## W

- WaitForChanged method, 732
- Watch window, 25, 26
- Watt, Allan H. (*3D Computer Graphics*), 762
- wavy underlines in code editor, 58–59

- WebBrowser control, 902–903
- When Breakpoint Is Hit dialog, 31–32**
- While End statement**
  - Do While statement and, 227, 793
  - enumerators, 223–224
  - iterators, 224–225
  - overview, 227, 792
- While loops**
  - Continue statement, 228
  - Do Loop, 225–227, 792
  - enumerators, 223–224
  - Exit statement, 227–228
  - iterators, 224–225
  - overview, 216
  - While End, 227, 792–793
- Window menu, 45–47**
- WindowProc (WndProc) routine, overriding, 285–289
- Windows Control Library template, 362**
- Windows forms controls. See also controls**
  - choosing, 264–272
  - container controls, 265–266
  - for displaying data, 268–269
  - for displaying dialogs, 271–272
  - for displaying graphics, 271
  - for entering data, 267–268
  - importance of, 259
  - for initiating action, 270–271
  - for making selections, 266–267
  - overview, 273–274
  - for providing feedback, 269–270
  - for supporting other controls, 272
  - table summarizing, 259–264
  - third-party controls, 272–273
- windows, rearranging, 48–49**
- Windows submenu (Debug menu), 26–30**
- Windows XP Control Panel, 677–678**
- WithEvents variables
  - catching events and, 74, 450
  - declaring, 143–145
  - event handlers, 92–93
- wizards, forms as, 303–304**
- WndProc routine, overriding, 285–289
- Word (Microsoft), printing from, 636**
- write-only property procedures, 788**

## X

- XML (Extensible Markup Language)**
  - comments, 122–125
  - overview, 745
  - System.Xml namespace for, 746–747

## Y

- Y2K problem, 233**