

Index

SYMBOLS

& (ampersand), shortcut indicator, 79

@ (at symbol), escape character, 108

A

Abort() method, 296

Accept() method, 198–199

AcceptSocket() method, 178–187

AcceptTcpClient() method, 178–187

AccessViolationException, 290

ActiveSync

bootstrapping, 460–461

delivering applications, 61

description, 42–43

adaptive networking, 6

Address property, 207–208

ADO.NET, 132–134

Alignment property, 398

AlphaABC input mode, 97

AlphaCurrent input mode, 97

AlphaT9 input mode, 97

Amazon Web Services, 228

ambiguous word input, 96–97

ampersand (&), shortcut indicator, 79

anonymous access, 353–354

antivirus measures, 331–332

Append() method, 27

application development

See *also* development tools

See *also* managed code

See *also* .Net Compact Framework

See *also* .Net Framework

See *also* unmanaged code

ActiveSync deployment, 61

adaptive networking, 6

automatic variables, 55

breakpoint configuration, 55

building applications, 47

CAB files, 56

challenges of, 6–7

choosing a platform, 17

connecting Smartphone devices, 18

creating projects, 43–49

debugging, 52–56

deploying applications, 61

deployment, 61

digital signatures, 59–61

e-mail attachment deployment, 61

Exceptions command, 55

form code, 49–51

form controls, 47

installing applications, 61

lightweight computation, 7

local variables, 55

online help, 18

packaging applications, 56–59

application development (continued)

- physical device emulation, 17–18
- power management, 7
- project files, 51
- Properties window, 46
- required procedures, 17–18
- security, 7
- security model, 59–60
- security policies, 59–60
- SI (Service Indication) messages, 61
- signing applications, 59–61
- SL (Service Loading) messages, 61
- solutions, 45
- stacking controls, 47
- starting without debugging, 48–49
- storage card deployment, 61
- storage efficiency, 6
- tab order, controls, 47
- templates, 44
- testing, 52–56
- user-friendly GUI, 6
- watched variables, 55
- web download deployment, 61

application domains, 13, 437

application signing, 309

`ApplicationException`, **290–292**

appointments. See PIM data.

`AppVerifier` tool, **295**

`ArgumentException`, **290**

arithmetic error exception, 290

`ArithmeticException`, **290**

`Array` class, **33–34**

`ArrayList` class, **34**

arrays

- C#, 33–34
- out of range exception, 290
- passing, 266–267

ASP.NET Mobile Controls, 16–17

assemblies, new features, 437

Assemblies namespace, 445

asymmetric cryptography, 335

asynchronous methods, 198–199

asynchronous web access, 170–171

at symbol (@), escape character, 108

“Attach to process” support, 293

attachments, filenames for, 210–211

`Attachments` property, **210–216**

attributes, .Net Compact Framework, 21

authentication

- information, getting, 168
- network, 346–347
- web services, 353–354
- Windows domain servers, 353–354

Auto Run Policy, 322

automatic scroll, 69–78

automatic variables, 55

autos window, 293

auto-save mode, 102

`AutoScaleMode` property, **102**

`AutoScroll` property, **69–78**

B

Back key, 9, 86–87

`BackgroundWorker` class, **430–431**

base classes, .Net Framework, 14

BCL collections, performance, 424–425

`BeginGetRequestStream()` method, **167–175**

`BeginGetResponse()` method, **167–175**

`BeginReceive()` method, **198–199**

`BeginSend()` method, **198–199**

`BeginUpdate()` method, **433–434**

big-endian byte order, 177–178

`BinaryReader` class, **110–112**

`BinarySearch()` method, **33–34**

`BinaryWriter` class, **110–112**

biometrics, 330

`BitArray` class, **34**

bitmaps, drawing, 402–408

blittable data types, 264–266, 419

blittable fields, passing, 268–270

blitting, 403

blocking mode, 198–199

bootstrapping, 459–464

breakpoint configuration, 55

breakpoint window, 293

breakpoints, 293
browsing files and directories, 113–120
Brush objects, 387–388
byte order conversion, 177–178

C

C#. See also .Net Compact Framework, type system.

Array class, 33–34
 ArrayList class, 34
 arrays, 33–34
 BinarySearch() method, 33–34
 BitArray class, 34
 CaseInsensitiveComparer class, 34
 casting data types, 23
 checked operator, 26
 classes, 28–32
 collections, 33–34
 Collections namespace, 34
 Comparer class, 34
 converting values, 23
 copying data, 24
 CreateInstance() method, 33–34
 data loss during conversion, 23
 enum data types, 24
 GetType() method, 26
 Hashtable class, 34
 ICollection interface, 34
 IComparer interface, 34
 IDictionary interface, 34
 IEnumerable interface, 34
 IEnumerator interface, 34
 IList interface, 34
 IndexOf() method, 33–34
 initializing values, 23
 Length property, 33–34
 object reference type, 26–27
 operators, 25–26
 OverflowException exception, 23
 polymorphism, 32–33
 Queue class, 34
 reference types, 24

shallow copy, 24
 Sort() method, 33–34
 SortedList class, 34
 Stack class, 34
 static methods, 23
 strings, 23–27
 struct data types, 24
 System.Convert class, 23
 typeof operator, 26
 unchecked operator, 26
 value types, 23–24
 virtual keyword, 32–33

CAB files, 56, 437

Cabir.A virus, 307–308

CABS Policy, 324

calendars, localization, 377–379. See also PIM data.

Call key, 9

call overhead, performance, 418–423

call stack window, 293

call-by-reference, 30

call-by-value, 30

calling conventions, P/Invoke, 260

card view, 87–88

CAS (Code Access Security), 308

case conversion, 27

CaseInsensitiveComparer **class, 34**

casting data types, 23

certificate stores, 310

certificates

definition, 309

loading, 321

MakeCert.exe tool, 319

managing, 319–321

obtaining, 314–315

Pvk2pfx.exe tool, 319

removing, 321

signing, 315–319

signtool utility, 319

System.Security.Cryptography.X509Certificates namespace, 453

viewing, 321

XML provisioning file, 321

chaining exceptions, 286

character coding, P/Invoke, 261

character encoding, localization, 381–382

checked **operator, 26**

class libraries

.Net Compact Framework, 15–16

.Net Framework, 13–14

performance, 424–428

class objects, 20

classes. See also C#, classes; interfaces;

namespaces.

accessibility, 32

Array, 33–34

ArrayList, 34

BackgroundWorker, 430–431

base, 14

BinaryReader, 110–112

BinaryWriter, 110–112

BitArray, 34

CaseInsensitiveComparer, 34

code-generation control, 14

Comparer, 34

CultureInfo, 365–366

CurrentCulture, 365–366

CurrentUICulture, 365–366

data access, 14

DataSet, 240–242

Directory, 107–108, 112

DirectoryInfo

directory-related operations, 107–108,
112

memo application, 120–128

EmailAccount, 206–208

EmailMessage, 206–208

Encoding, 381–382

Exception, 288–289

File, 107–108, 112

FileInfo

file-related operations, 107–108, 112
memo application, 120–128

FileStream, 108–110, 112

FileStreamInfo, 112

FontCollection, 386

GenericFontFamilies, 386

Graphics, 386

Hashtable, 34

HttpRequest, 168–169

HttpResponse, 168–169

InputModeEditor, 97

InstalledFontCollection, 386

I/O, 14

IOException, 112

IPAddress, 176–177

IPEndPoint, 176

IPHostEntry, 176–177

language compilation, 14

MemoryStream, 108–110, 112

MessageInterceptor, 221–224

MessagingApplication, 211–216

networking, 14

OutlookSession, 204–212, 218–224

passing, 268–270, 270–271

Path, 112

PimItemCollection, 219–220

PrivateFontCollection, 386

Queue, 34

Recipient, 206–208

Regex, 427–428

Region, 389–391

remoting, 14

security, 14

SmsMessage, 221–225

Socket, 187–188

SortedList, 34

Stack, 34

Stream, 108–110, 112

StreamReader, 111–112

StreamWriter, 111–112

String, 426–427

StringBuilder, 27, 267–268, 426–427

StringReader, 112

StringWriter, 113

System.Convert, 23

SystemException, 288–289

System.Object, 18

TcpClient, 178–187

TcpListener, 178–187

TextReader, 110–112, 113

TextWriter, 110–112, 113

- threading, 14
- UI, 14
- web application, 14
- web services, 14
- WebRequest, 167–175
- WebResponse, 167–175
- XmlDocument, 235–240
- XmlNodeReader, 235, 240–242
- XmlNodeWriter, 235
- XmlReader, 235, 428
- XMLSerializer, 429
- XmlTextReader, 235–240
- XmlTextWriter, 235
- XmlWriter, 235, 428
- CLI (common language infrastructure), 12**
- client environment, SQL Server 2005**
 - Mobile, 136. See also TCP servers and clients.**
- clipping, 392–393**
- CLR (Common Language Runtime). See also performance, CLR.**
 - application domain, 13
 - CTS (common type system), 13
 - .Net Compact Framework, 15
 - .Net languages, 13
 - services provided, 12–13
- CLR host, 13**
- Code Access Security (CAS), 308**
- code-generation control classes, 14**
- collections**
 - iterating through, performance, 425
 - overview, 33–34
 - System.Collections namespace, 443
 - System.Collections.Generic namespace, 443
 - System.Collections.ObjectModel namespace, 443
 - System.Collections.Specialized namespace, 443–444
- Collections namespace, 34, 443**
- Collections.Generic namespace, 443**
- Collections.ObjectModel namespace, 443**
- Collections.Specialized namespace, 443–444**
- Color objects, 387–388**
- color transparency, 404**
- COM calls, performance, 418–420**
- COM interop, 16**
- command mode, 293**
- commerce, smartphone uses for, 5**
- common language infrastructure (CLI), 12**
- Common Language Runtime (CLR). See CLR (Common Language Runtime).**
- Common namespace, 446**
- common type system (CTS), 13**
- communication security. See security, communications.**
- Comparer class, 34**
- comparing strings, 27**
- compilers, .Net Framework, 13**
- CompilerServices namespace, 452**
- ComponentModel namespace, 445**
- concatenating strings, 27**
- condition checking versus exceptions, 292**
- Configuration.Assemblies namespace, 445**
- Connect () method, 192–196, 198–199**
- connected mode, 132–133**
- consuming web services, 249–252**
- contacts. See PIM data.**
- ContentLength property, 168**
- ContentType property, 168**
- controls, hardware**
 - Back key, 86–87
 - Home key, 86–87
 - Microsoft design guidelines, 85
 - soft keys, 86, 101–102
- controls, software**
 - automatic scroll, 69–78
 - available, list of, 64–65
 - focus behavior, 74–78, 84–85
 - ListView, 84–85
 - MainMenu, 78–84
 - menus, 78–84
 - Panel, 84–85
 - repainting, performance, 433–434
 - submenus, 78–79
 - tab order, 66–69
 - TreeView, 84–85

Convert class

Convert **class**, 23

Convert() **method**, 381–382

converting strings to numbers, 23

converting values, C#, 23

Coordinated Universal Time (UTC), 377

copying

data, C#, 24

directories, 107–108

files, 107–108

counters, performance, 412–417

Create() **method**

File class, 107–108

files, creating, 107–108

web requests, creating, 169

WebRequest class, 169

CreateGraphics() **method**, 387

CreateInstance() **method**, 33–34

creating

databases, 138–142, 147–148, 157–158

directories, 107–108

e-mail applications

with attachments, 210–216

with managed APIs, 205–208

simple version, 208–210

files, 107–108

Graphics objects, 387

menus, 78–84

projects, 43–49

publications, 142–145

subscriptions, 148–151

tables, 138–142

threads, 297

web requests

asynchronous access, 170–171

Create() method, 169

example, 171–175

overview, 169

synchronous access, 169–170

Credentials **property**, 168

critical sections, 298

cryptography, 334–335, 437. *See also*
encryption.

Cryptography.X509Certificates namespace,
453

CTS (common type system), 13

cultural influences. *See* globalization;
localization.

culture codes, 364–365

CultureInfo **class**, 365–366

currency, localization, 379–381

CurrentCulture **class**, 365–366

CurrentThread() **method**, 296

CurrentUICulture **class**, 365–366

D

data. *See also* security, data.

access classes, 14

encryption, 334–341

loss during conversion, 23

providers, 132

service, smartphone uses for, 6

Data namespace, 445–446

data storage

object store, 106

overview, 105–107

persistent storage filesystems, 106–107

program memory, 106

RAM-based filesystems, 106

unified directory view, 107

data types. *See also* .Net Compact
Framework, type system.

blittable, 264–266

.Net Compact Framework, 19

.Net Framework, 13

XML, 234

data types, C#. *See also* .Net Compact
Framework, type system.

casting, 23

enum, 24

GetType() method, 26

object reference type, 26–27

reference types, 24

struct data types, 24

typeof operator, 26

value types, 23–24

databases

creating, 138–142, 147–148, 157–158

encryption, 341–346

databases, relational data. See also SQL Server 2005 Mobile.

- ADO.NET, 132–134
- connected mode, 132–133
- converting from XML, 240–242
- data providers, 132
- disconnected mode, 132, 133–134

Data.Common namespace, 446

`DataGrid` control, example, 148–151

`DataSet` class, 240–242

`Data.SqlClient` namespace, 446–447

`Data.SqlServerCe` namespace, 447

`Data.SqlTypes` namespace, 447–448

dates, localization, 377–379**deadlock, 300–304**

`/debug` option, C#, 294

debugging

- auto variables, 55
- breakpoint configuration, 54
- controls, 54
- description, 52–56
- Edit and Continue feature, 55
- limitations, 55–56
- local variables, 55
- selecting exceptions to monitor, 55
- starting without, 48–49
- `System.Diagnostics` namespace, 448
- watched variables, 55

debugging, with Visual Studio 2005

- `AppVerifier` tool, 295
- “Attach to process” support, 293
- autos window, 293
- breakpoint window, 293
- breakpoints, 293
- call stack window, 293
- command mode, 293
- `/debug` option, C#, 294
- Edit-and-Continue, 295
- expressions, entering, 293
- immediate mode, 293
- immediate window, 293
- Just-My-Code debugging, 295
- limitations, 295
- loaded modules, listing, 293
- locals window, 293

- method call stack, displaying, 293
- modules window, 293
- multithreaded
 - `Abort()` method, 296
 - background threads, 297
 - blocking a calling thread, 296
 - creating a thread, 297
 - critical sections, 298
 - current thread, returning, 296
 - current thread, sleeping, 296
 - `CurrentThread()` method, 296
 - deadlock, 300–304
 - `IsBackground()` method, 297
 - `Join()` method, 296
 - locks, 298–300
 - managed threads, 296–297
 - `ManagedThreadId()` method, 296
 - `Name()` method, 296
 - overview, 296
 - `Priority()` method, 297
 - race conditions, 298–300
 - `Sleep()` method, 296
 - `Start()` method, 296
 - starting a thread, 296–297
 - terminating a thread, 296
 - thread ID, returning, 296
 - thread names, getting/setting, 296
 - thread pool, 297
 - thread priority, 297
- next statement, 295
- objects, displaying, 293
- process window, 293
- running processes, listing, 293
- settings, 294
- statements, entering, 293
- symbols, defining, 294–295
- threads, listing, 293
- threads window, 293
- variables, displaying, 293
- viewing properties and fields, 293
- watch windows, 293
- windows, 293–294
- `Default` input mode, 97
- delegate keyword, 30–31
- delegates, 30–31**

deleting files and directories, 107–108

delivering applications. See deploying applications.

deploying applications, 61

depth-first node reading, 236

Description namespace, 455

developing applications. See application development.

development environment, 135, 136

development tools. See also .Net Compact Framework; .Net Framework.

ActiveSync, 18, 42–43

all-in-one package, 18

AppVerifier, 295

bootstrapping, 460–461

certificates, 319

debugging, with Visual Studio 2005, 295

Device Emulator, 41–42

eMbedded C++, 17, 18

eMbedded Visual Basic, 17

MakeCert.exe, 319

MSBuild, 40

.Net Remote, 40

Pvk2pfx.exe, 319

RapiConfig tool, 460–461

signtool, 319

SQL Server 2005 Mobile, 138

Visual Studio 2005, 17–18, 40–41

Windows Mobile 5.0, 7–8, 41

Device Emulator, 41–42

Diagnostics namespace, 448

DiffGram mode, 242

digest authentication, 353–354

digital signatures, 59–61, 309

Direct3D support, 437

directories. See also DirectoryInfo class.

browsing, 113–120

copying, 107–108

creating, 107–108

getting information about, 107–108

moving, 107–108

Directory class, 107–108, 112

DirectoryInfo class

directory-related operations, 107–108, 112

memo application, 120–128

DirectX support, 437

disconnected mode, 132, 133–134

DisplayComposeForm() method, 211–216

document elements, 231

DotGNU project, 12

double buffering, 403

DrawEllipse() method, 393–397

DrawGrid() method, 394–397

drawing. See also graphics.

bitmaps, 402–408

ellipses, 393–397

grids, 394–397

images, 400–408

lines, 387–388, 393–397

polygons, 393–397

rectangles, 393–397

shapes, 393–397

strings, 397–398

text, 397–398

Drawing namespace, 448

Drawing2D namespace, 449

Drawing.Imaging namespace, 449

Drawing.Text namespace, 449

DrawLine() method, 393–397

DrawLines() method, 393–397

DrawPolygon() method, 393–397

DrawRectangle() method, 393–397

DrawString() methods, 397–398

Dumpbin.exe utility, 258–260

E

Earpiece volume, 9

eBay Developers Program, 228

Edit-and-Continue, 295

ellipses, drawing, 393–397

e-mail

deploying applications with, 61

SMS, 220–225

text messaging, 220–225

e-mail applications

accounts, 206–208

Address property, 207–208

Attachments property, 210–216

- creating
 - with attachments, 210–216
 - with managed APIs, 205–208
 - simple version, 208–210
- DisplayComposeForm() method, 211–216
- EmailAccount class, 206–208
- EmailMessage class, 206–208
- filenames for attachments, 210–211
- messages, 207–208
- MessagingApplication class, 211–216
- Name property, 207–208
- OutlookSession class, 204–212, 218–224
- POOM (Pocket Outlook Object Model), 202–203
- receiving messages, 206–208
- Recipient class, 206–208
- Send() method
 - EmailAccount class, 208
 - SmsAccount class, 221–225
- sending messages, 208
- WindowsMobile.PocketOutlook namespace, 203–205
- EmailAccount **class, 206–208**
- EmailMessage **class, 206–208**
- eMbedded C++, 17, 18**
- embedded graphics resources, 408–409**
- eMbedded Visual Basic, 17**
- emulator networking, 166–167**
- Encoding **class, 381–382**
- encryption**
 - asymmetric cryptography, 335
 - cryptography algorithms, 334–335
 - data, 334–341
 - databases, 341–346
 - example, 335–341
 - hardware, 334
 - hashing cryptography, 335
 - symmetric cryptography, 334
- EndGetRequestStream() **method, 167–175**
- EndGetResponseStream() **method, 167–175**
- EndReceive() **method, 198–199**
- EndSend() **method, 198–199**
- EndUpdate() **method, 433–434**
- enterprise, smartphone uses for, 5**
- entertainment, smartphone uses for, 6**
- enum **data types, 24**
- environment variables, Windows, 258–260**
- error handling. See also exception handling; exceptions.**
 - MissingMethodException, 263
 - NotSupportedException, 263
 - P/Invoke, 262–263
- events, C#, 30–31**
- Exception **class, 288–289**
- exception classes versus user-defined exception classes, 292**
- exception handling. See also error handling.**
 - accessing protected memory, 290
 - AccessViolationException, 290
 - ApplicationException, 290–292
 - ArgumentException, 290
 - arithmetic errors, 290
 - ArithmeticException, 290
 - array, out of range, 290
 - base class, 288–289
 - best practices, 292
 - Exception class, 288–289
 - finally statement, 283–284
 - GetBaseException() method, 289
 - HResult property, 289
 - IndexOutOfRangeException, 290
 - InnerException property, 289
 - input/output, 290
 - invalid method parameter, 290
 - IOException, 290
 - Message property, 289
 - .Net Compact Framework, 21–22
 - NotImplementedException, 290
 - NotSupportedException, 290
 - null object access, 290
 - NullReferenceException, 290
 - out of memory, 290
 - OutOfMemoryException, 290
 - purpose of, 282
 - request method does not exist, 290
 - request method not implemented, 290

exception handling (continued)

- security errors, 290
- SecurityException, 290
- StackTrace property, 289
- SystemException class, 288–289
- throw statement, 285–286
- ToString() method, 289
- try...catch block, 110
- try...catch statement, 282–283
- XML processing errors, 290
- XmlException, 290

exception stack trace, 286–288

exceptions. See also error handling.

- chaining, 286
- versus condition checking, 292
- converting to strings, 289
- definition, 282
- handled, 282
- P/Invoke, 261
- versus return code error, 292
- root cause, 289
- unhandled, 282, 289–290
- wrapping, 286

Exceptions command, 55

exceptions counters, 416

exporting DLL functions, 261–262

expressions, entering in debugger, 293

eXtensible Markup Language (XML). See XML (eXtensible Markup Language).

extern keyword, 262

F

fields, viewing, 293

File class, 107–108, 112

file manipulation classes, 107–108

FileInfo class

- file-related operations, 107–108, 112
- memo application, 120–128

filenames for attachments, 210–211

files. See also FileInfo class.

- accessing, 108–110
- attaching to e-mail, 210–211
- browsing, 113–120

- copying, 107–108

- creating, 107–108

- moving, 107–108

- reading and writing, 108–112

FileStream class, 108–110, 112

FileStreamInfo class, 112

filesystems, security, 308

FillEllipse() method, 394–397

FillPolygon() method, 394–397

FillRectangles() method, 394–397

FillRegion() method, 394–397

fills, 394–397

finalization queue, 20

finalizer, 20

finally statement, 283–284

focus behavior, 74–78, 84–85

FontCollection class, 386

fonts, 399–400

foreach() method, 425

fork dinner problem, 300

FormatOutput() method, 238–240

formatting XML output, 238–240

forms. See also performance, forms.

- code, 49–51

- controls, 47

- drawing grids on, 394–397

- Microsoft.WindowsCE.Forms namespace, 442

- multiple, 88–96

- performance, 103

- System.Windows.Forms namespace, 455–456

- Windows Forms control, new features, 439–440

- WindowsCE Forms control, new features, 439

Forms namespace, 455–456

FromImage() method, 387, 402–403

G

garbage collection

- counters, 416

- .Net Compact Framework, 20

- performance, 417–418

Gates, Bill, 4**Generic namespace, 443**GenericFontFamilies **class, 386****generics**

counters, 416

.Net Compact Framework, 21

new features, 437

performance, 423–424

GetBaseException() **method, 289**GetFocus **event handler, 74**GetLastError **property, 263**GetPixel() **method, 405–408**GetRequestStream() **method, 167–175**GetResponse() **method, 167–175**GetType() **method, 26****globalization. See also localization.**

cultural influences, 364–366

culture codes, 364–365

CultureInfo class, 365–366

CurrentCulture class, 365–366

CurrentUICulture class, 365–366

definition, 363

MUI (Multilingual Language Interface),
365–366

new features, 437

world-ready applications

localized resources, 366–369

localized satellite assembly, 370–377

sample, 370–377

Globalization namespace, 449**GMT (Greenwich Mean Time), 377****Google Web APIs, 228****Grant Manager Policy, 322****Grant User Authenticated Policy, 322****graphics**

new features, 438

System.Drawing namespace, 448

System.Drawing.Drawing2D namespace,
449

System.Drawing.Imaging namespace, 449

System.Drawing.Text namespace, 449

text

System.Text namespace, 453

System.Text.RegularExpressions
namespace, 454**graphics, .Net Compact Framework**

Alignment property, 398

blitting, 403

Brush objects, 387–388

clipping, 392–393

Color objects, 387–388

color transparency, 404

CreateGraphics() method, 387

double buffering, 403

DrawEllipse() method, 393–397

DrawGrid() method, 394–397

drawing

bitmaps, 402–408

images, 400–408

lines, 387–388

shapes, 393–397

text, 397–398

DrawLine() method, 393–397

DrawLines() method, 393–397

DrawPolygon() method, 393–397

DrawRectangle() method, 393–397

DrawString() methods, 397–398

embedded resources, 408–409

FillEllipse() method, 394–397

FillPolygon() method, 394–397

FillRectangles() method, 394–397

FillRegion() method, 394–397

fills, 394–397

FontCollection class, 386

fonts, 399–400

FromImage() method, 387, 402–403

GenericFontFamilies class, 386

GetPixel() method, 405–408

Graphics class, 386

Graphics objects, creating, 387

grids on forms, 394–397

InstalledFontCollection class, 386

IsInvisible() method, 389

LineAlignment property, 398

MeasureString() method, 398

PaintEventArgs control, 387

painting, 387–388

Pen objects, 387–388

PrivateFontCollection class, 386

Region class, 389–391

graphics, .Net Compact Framework (continued)

- regions, 389–391
- releasing resources, 387
- SetPixel() method, 405–408
- text alignment, 398
- text wrap, 398
- types of classes, 386
- vector graphics, 389–397
- vector methods, 393–397
- Graphics **class**, 386
- Graphics **objects, creating**, 387
- Greenwich Mean Time (GMT), 377
- grids, drawing, 394–397
- GUI-related counters, 417

H

- handled exceptions, 282
- Hang-up key, 9
- hardware
 - controls
 - Back key, 86–87
 - Home key, 86–87
 - Microsoft design guidelines, 85
 - soft keys, 86, 101–102
 - encryption, 334
- hash codes, 309
- hashing cryptography, 335
- Hashtable **class**, 34
- Home key, 9, 86–87
- HResult **property**, 289
- HTTP version, getting, 168
- HttpRequest **class**, 168–169
- HttpResponse **class**, 168–169

I

- ICollection **interface**, 34
- IComparer **interface**, 34
- IDictionary **interface**, 34
- IEnumerable **interface**, 34
- IEnumerator **interface**, 34

- IgnoreSchema **mode**, 242
- IIS (Internet Information Services), 136, 353–354
- IList **interface**, 34
- images, drawing, 400–408. *See also* graphics.
- Imaging namespace, 449
- immediate mode, 293
- immediate window, 293
- IndexOf() **method**, 33–34
- IndexOutOfRangeException, 290
- inheritance, multiple, 31
- InitializeComponent() **method**, 431–433
- initializing values, 23
- inline schemas, 234
- InnerException **property**, 289
- input mode, 96–101
- InputMode **enumeration**, 97
- InputModeEditor **class**, 97
- input/output. *See also* I/O classes; System.IO namespace.
 - exceptions, 290
 - file access, 108–110
 - System.IO namespace, 450
 - System.IO.Ports namespace, 450
 - text, 110–112
- Insert() **method**, 27
- InstalledFontCollection **class**, 386
- installing
 - applications, 61
 - SQL Server 2005 Mobile
 - client environment, 137, 138
 - development environment, 137
 - server environment, 137
 - system requirements, 136–137
- instances. *See* reference types.
- Integrated Windows authentication, 353–354
- interfaces. *See also* classes.
 - C#, 31
 - ICollection, 34
 - IComparer, 34
 - IDictionary, 34
 - IEnumerable, 34
 - IEnumerator, 34
 - IList, 34

internal modifier, **31**
 internal protected modifier, **31**
Internet Information Services (IIS), 136, 353–354
 interop counters, **416**
 interoperability, new features, **438**
InteropServices namespace, 452–453
I/O. See input/output; System.IO namespace.

I/O classes

byte-level, 108–110
 higher-level, 110–112
 .Net Framework, 14
IOException, 290
IOException class, 112
IO.Ports namespace, 450
IP endpoints, 176–177
IPAddress class, 176–177
IPEndPoint class, 176
IPHostEntry class, 176–177
IPSec (IP Security), 351
IsBackground() method, 297
IsInvisible() method, 389

J

JIT compilation counters, 416
Join() method, 296
Just-My-Code debugging, 295

K

Kerberos, 347
keyboard events, new features, 438
keyboard input. See user input.

L

L2TP (Layer 2 Tunneling Protocol), 351
language compilation classes, 14
layout management, new features, 438

Length property, 33–34
lightweight computation, 7
LineAlignment property, 398
lines, drawing, 387–388, 393–397
list view, 87–88
ListView, 84–85
literal strings, 27
little-endian byte order, 177–178
Load() method, 235–240
loaded modules, listing, 293
loader counters, 416
loading XML files, 235–240
LoadXml() method, 240
local variables, 55
localization. See also globalization.
 best practices, 382–383
 calendars, 377–379
 character encoding, 381–382
 Convert() method, 381–382
 cultural influences, 364–366
 culture codes, 364–365
 CultureInfo class, 365–366
 currency, 379–381
 CurrentCulture class, 365–366
 CurrentUICulture class, 365–366
 dates, 377–379
 definition, 363–364
 Encoding class, 381–382
 GMT (Greenwich Mean Time), 377
 MUI (Multilingual Language Interface), 365–366
 numbers, 379–381
 strings, 381–382
 time, 377–379
 time zones, 377–379
 UTC (Coordinated Universal Time), 377
 world-ready applications
 localized resources, 366–369
 localized satellite assembly, 370–377
 sample, 370–377
locals window, 293
Locked setting, 314
locks, 298–300
loss, 308

M

mail. See **e-mail.**

MainMenu control, 78–84

managed code

converting to unmanaged, 263–271

definition, 11

exporting functions from, 262

memory management, 256

portability, 256

reusability, 256

versus unmanaged, 255–257

managed DLLs, 257–260

`ManagedThreadId()` **method, 296**

`MarshalAs` **attribute, 270–271**

marshaling data

arrays, passing, 266–267

blittable fields, passing, 268–270

blittable types, 264–266

classes, passing, 268–270, 270–271

data fields, physical layout, 269–270

definition, 263

`MarshalAs` attribute, 270–271

non-blittable fields, passing, 270–271

out parameter, 265–266

passing by reference, 265–266

ref parameter, 265–266

reference types, 266–268

string keyword, 267–268

string variables, passing, 267–268

`StringBuilder` class, 267–268

structures, passing, 268–270, 270–271

`UnmanagedType` enumeration, 270–271

value types, 264–266

math operations, performance, 423

`MeasureString()` **method, 398**

member accessibility, C# classes, 31

members, C# classes, 28–29

memo application, 120–128

memory

out of memory exception, 290

usage counters, 416

`MemoryStream` **class, 108–110, 112**

MenuItem controls, 78–84

menus

creating, 78–84

MainMenu control, 78–84

MenuItem controls, 78–84

shortcut keys, 78–79

submenus, 78–79

Merge publication, 144–145

Message Authentication Retry Number Policy, 323

message digests, 309

Message **property, 289**

message queuing, new features, 438

`MessageInterceptor` **class, 221–224**

messages, e-mail. See e-mail.

Messaging namespace, 450–451

`MessagingApplication` **class, 211–216**

method call stack, displaying, 293

Method **property, 168**

methods

`Abort()`, 296

`Accept()`, 198–199

`AcceptSocket()`, 178–187

`AcceptTcpClient()`, 178–187

`Append()`, 27

asynchronous, 198–199

`BeginGetRequestStream()`, 167–175

`BeginGetResponse()`, 167–175

`BeginReceive()`, 198–199

`BeginSend()`, 198–199

`BeginUpdate()`, 433–434

`BinarySearch()`, 33–34

call stack, displaying, 293

`Connect()`, 192–196, 198–199

`Convert()`, 381–382

`Create()`

File class, 107–108

files, creating, 107–108

web requests, creating, 169

`WebRequest` class, 169

`CreateGraphics()`, 387

`CreateInstance()`, 33–34

`CurrentThread()`, 296

`DisplayComposeForm()`, 211–216

`DrawEllipse()`, 393–397

- DrawGrid(), 394–397
- DrawLine(), 393–397
- DrawLines(), 393–397
- DrawPolygon(), 393–397
- DrawRectangle(), 393–397
- EndGetRequestStream(), 167–175
- EndGetResponseStream(), 167–175
- EndReceive(), 198–199
- EndSend(), 198–199
- EndUpdate(), 433–434
- FillEllipse(), 394–397
- FillPolygon(), 394–397
- FillRectangles(), 394–397
- FillRegion(), 394–397
- foreach(), 425
- FormatOutput(), 238–240
- FromImage(), 387, 402–403
- GetBaseException(), 289
- GetPixel(), 405–408
- GetRequestStream(), 167–175
- GetResponse(), 167–175
- GetType(), 26
- IndexOf(), 33–34
- InitializeComponent(), 431–433
- Insert(), 27
- invalid parameter exception, 290
- IsBackground(), 297
- IsInvisible(), 389
- Join(), 296
- Load(), 235–240
- LoadXml(), 240
- ManagedThreadId(), 296
- MeasureString(), 398
- MoveToAttribute(), 236–237
- MoveToContent(), 236–240
- Name(), 296
- Pending(), 179–187
- Prelink(), 419–420
- PrelinkAll(), 419–420
- Priority(), 297
- ProcessConnection(), 189–192
- Read(), 178–187, 236–240
- ReadXml(), 241–242
- Receive(), 189–196, 198–199
- ReceiveFrom(), 198–199
- Remove(), 27
- request method does not exist, 290
- request method not implemented, 290
- Restrict(), 219–220
- ResumeLayout(), 431–433
- SelectNode(), 240
- SelectSingleNode(), 240
- Send()
 - EmailAccount class, 208
 - sending e-mail, 208
 - SmsAccount class, 221–225
 - sockets, blocking mode, 198–199
 - TCP client sockets, 192–196
 - TCP server sockets, 189–192
- SendTo(), 198–199
- SetPixel(), 405–408
- ShowResults(), 252
- ShutdownServer(), 189–192
- Skip(), 236–237
- Sleep(), 296
- Sort(), 33–34
- Start(), 296
- StartServer(), 189–192
- static, C#, 23
- SuspendLayout(), 431–433
- ToLower(), 27
- ToString(), 23, 27, 289
- ToUpper(), 27
- WorkerThreadProc(), 179–187
- Microsoft SQL Server 2005 Mobile Edition.**
See SQL Server 2005 Mobile.
- Microsoft Windows. See Windows.**
- Microsoft.VisualBasic namespace, 441**
- Microsoft.VisualBasic.CompilerServices namespace, 441–442**
- Microsoft.WindowsCE.Forms namespace, 442**
- MissingMethodException, 263
- mobile devices**
 - memory size, 15
 - software emulation, 17–18
- modal dialog, 91**
- modeless dialog, 91**
- modules window, 293**
- Mono project, 12**

MoveToAttribute() method

`MoveToAttribute()` method, 236–237
`MoveToContent()` method, 236–240
moving files and directories, 107–108
MSBuild, 40
MSIL (Microsoft Intermediate Language), 13
MSN Search, 228
MUI (Multilingual Language Interface), 365–366
multiple inheritance, C# classes, 31
multi-tap function, 96–101
multithreaded debugging
 `Abort()` method, 296
 background threads, 297
 blocking a calling thread, 296
 creating a thread, 297
 critical sections, 298
 current thread, returning, 296
 current thread, sleeping, 296
 `CurrentThread()` method, 296
 deadlock, 300–304
 `IsBackground()` method, 297
 `Join()` method, 296
 locks, 298–300
 managed threads, 296–297
 `ManagedThreadId()` method, 296
 `Name()` method, 296
 overview, 296
 `Priority()` method, 297
 race conditions, 298–300
 `Sleep()` method, 296
 `Start()` method, 296
 starting a thread, 296–297
 terminating a thread, 296
 thread ID, returning, 296
 thread names, getting/setting, 296
 thread pool, 297
 thread priority, 297

N

name mangling, 258
`Name()` method, 296
`Name` property, 207–208

namespaces. See also classes; .Net Compact Framework, class library.
`Microsoft.VisualBasic`, 441
`Microsoft.VisualBasic.CompilerServices`, 441–442
`Microsoft.WindowsCE.Forms`, 442
`SqlServerCe`, 157–163
`System`, 442–443
`System.IO`
 `BinaryReader` class, 110–112
 `BinaryWriter` class, 110–112
 byte-level I/O classes, 108–110
 classes, summary of, 112–113
 classes, unsupported, 113
 description, 450
 `Directory` class, 107–108, 112
 `DirectoryInfo` class, 107–108, 112
 file access, 108–110
 `File` class, 107–108, 112
 file manipulation classes, 107–108
 `FileInfo` class, 107–108, 112
 `FileStream` class, 108–110, 112
 `FileStreamInfo` class, 112
 higher-level I/O classes, 110–112
 `IOException` class, 112
 `MemoryStream` class, 108–110, 112
 `Path` class, 112
 static classes, 107
 `Stream` class, 108–110, 112
 `StreamReader` class, 111–112
 `StreamWriter` class, 111–112
 `StringReader` class, 112
 `StringWriter` class, 113
 structures, unsupported, 113
 `TextReader` class, 110–112, 113
 `TextWriter` class, 110–112, 113
`WindowsMobile.PocketOutlook`, 203–205
native code. See unmanaged code.
.Net Compact Framework. See also graphics, .Net Compact Framework.
`ASPNET Mobile Controls`, 16–17
class libraries, 15–16
CLR (Common Language Runtime), 15

- COM interop, 16
- counters, 412–417
- limitations, 308
- P/Invoke, 16–17
- security limitations, 308
- support for XML, 229
- unmanaged code, accessing, 16–17
- .Net Compact Framework, class library**
 - Microsoft.VisualBasic, 441
 - Microsoft.VisualBasic.CompilerServices, 441–442
 - Microsoft.WindowsCE.Forms, 442
 - System, 442–443
 - System.Collections, 443
 - System.Collections.Generic, 443
 - System.Collections.ObjectModel, 443
 - System.Collections.Specialized, 443–444
 - System.ComponentModel, 445
 - System.Configuration.Assemblies, 445
 - System.Data, 445–446
 - System.Data.Common, 446
 - System.Data.SqlClient, 446–447
 - System.Data.SqlServerCe, 447
 - System.Data.SqlTypes, 447–448
 - System.Diagnostics, 448
 - System.Drawing, 448
 - System.Drawing.Drawing2D, 449
 - System.Drawing.Imaging, 449
 - System.Drawing.Text, 449
 - System.Globalizatiion, 449
 - System.IO, 450
 - System.IO.Ports, 450
 - System.Messaging, 450–451
 - System.Net, 451
 - System.Net.Sockets, 451–452
 - System.Reflection, 452
 - System.Resources, 452
 - System.Runtime.CompilerServices, 452
 - System.Runtime.InteropServices, 452–453
 - System.Security, 453
 - System.Security.Cryptography.X509Certificates, 453
 - System.Security.Policy, 453
 - System.Text, 453
 - System.Text.RegularExpressions, 454
 - System.Threading, 454–455
 - System.Web.Services, 455
 - System.Web.Services.Description, 455
 - System.Windows.Forms, 455–456
- .Net Compact Framework, new features**
 - application domains, 437
 - assemblies, 437
 - CAB file installation, 437
 - cryptography, 437
 - Direct3D support, 437
 - DirectX support, 437
 - forms, 439–440
 - generics, 437
 - globalization, 437
 - graphics, 438
 - interoperability, 438
 - keyboard events, 438
 - layout management, 438
 - message queuing, 438
 - networking, 438
 - partial classes, 438
 - Registry keys, 438
 - Remote Performance Monitor, 439
 - serial ports, 439
 - SQL Server Mobile, 439
 - threads support, 439
 - Windows Forms control, 439–440
 - WindowsCE Forms control, 439
 - WindowsCE Forms control, new features, 439
 - XML support, 439–440
- .Net Compact Framework, type system. See also C#.**
 - attributes, 21
 - class objects, 20
 - data types, 19
 - exception handling, 21–22
 - finalization queue, 20
 - finalizer, 20
 - garbage collection, 20
 - generics, 21
 - numeric data types, 19
 - parametric polymorphism, 21
 - reference type objects, 19
 - reflection, 21

.Net Compact Framework, type system (continued)

- runtime inspection, 21
- System.Object class, 18
- types of objects, 19–20
- value type objects, 19

.Net Framework

- application domain, 13
- base classes, 14
- class libraries, 13–14
- CLI (common language infrastructure), 12
- CLR (Common Language Runtime), 12–13
- CLR host, 13
- code-generation control classes, 14
- compilers, 13
- components, 12
- CTS (common type system), 13
- data access classes, 14
- DotGNU project, 12
- enabling for Linux, 12
- introduction, 11–12
- I/O classes, 14
- language compilation classes, 14
- Mono project, 12
- MSIL (Microsoft Intermediate Language), 13
- .Net languages, 13
- networking classes, 14
- remoting classes, 14
- Rotor, 12
- runtime, 13
- security classes, 14
- threading classes, 14
- type, 13
- UI classes, 14
- virtual machine, 12–13
- Visual Studio 2005, 14
- web application classes, 14
- web services classes, 14

.Net languages, 13

Net namespace, 451

.Net Remote tools, 40

Net.Sockets namespace, 451–452

network authentication, 346–347

networking

- authentication information, getting, 168
- BeginGetRequestStream() method, 167–175
- BeginGetResponse() method, 167–175
- classes, 14
- ContentLength property, 168
- ContentType property, 168
- counters, 416
- Credentials property, 168
- emulator, 166–167
- EndGetRequestStream() method, 167–175
- EndGetResponseStream() method, 167–175
- GetRequestStream() method, 167–175
- GetResponse() method, 167–175
- HTTP version, getting, 168
- HttpRequest class, 168–169
- HttpResponse class, 168–169
- Method property, 168
- new features, 438
- overview, 165–166
- ProtocolVersion property, 168
- Proxy property, 168
- smartphone access, checking, 167
- sockets
 - Accept() method, 198–199
 - asynchronous methods, 198–199
 - BeginReceive() method, 198–199
 - BeginSend() method, 198–199
 - blocking mode, 198–199
 - Connect() method, 192–196, 198–199
 - EndReceive() method, 198–199
 - EndSend() method, 198–199
 - nonblocking mode, 198–199
 - overview, 187–188
 - ProcessConnection() method, 189–192
 - Receive() method, 189–196, 198–199
 - ReceiveFrom() method, 198–199

Send() method, 189–196, 198–199
 SendTo() method, 198–199
 ShutdownServer() method, 189–192
 Socket class, 187–188
 StartServer() method, 189–192
 TCP client, 192–196
 TCP server, 188–192
 UDP, 196–198
 System.Net namespace, 451
 System.Net.Sockets namespace, 451–452
 TCP servers and clients
 AcceptSocket() method, 178–187
 AcceptTcpClient() method, 178–187
 big-endian byte order, 177–178
 byte order conversion, 177–178
 IP endpoints, 176–177
 IPAddress class, 176–177
 IPEndPoint class, 176
 IPHostEntry class, 176–177
 little-endian byte order, 177–178
 Pending() method, 179–187
 Read() method, 178–187
 TCP client, 180–182
 TCP server, 182–187
 TcpClient class, 178–187
 TcpListener class, 178–187
 WorkerThreadProc() method, 179–187
 Timeout property, 168
 timeout value, 168
 web proxy, 168
 web requests, creating
 asynchronous access, 170–171
 Create() method, 169
 example, 171–175
 overview, 169
 synchronous access, 169–170
 WebRequest class, 167–175
 WebResponse class, 167–175
 new **keyword**, 30–31
 next **statement**, 295
non-blittable fields, passing, 270–271
nonblocking mode, 198–199
normal applications, 310–311

NotImplementedException, 290
 NotSupportedException, 263, 290
 NotSupportedException **error handler**, 263
NTLM (NT LAN Manager), 346–347
null object exception, 290
 NullReferenceException, 290
numbers, localization, 379–381
numeric data types, 19
 Numeric **input mode**, 97

O

Object **class**, 18
 object **reference type**, 26–27
object store, 106
ObjectModel namespace, 443
objects. See also reference types.
 displaying, 293
 types of, 19–20
one-tiered security, 312
One-Tier-Prompt setting, 313
online help, 18
operating systems
 Palm OS (Palm Inc.), 6
 for Pocket PCs, 7–8
 Symbian OS (Symbian), 6
 Windows Mobile 5.0, 7–8
operators, C#, 25–26
 out **parameter**, 265–266
Outlook API. See POOM (Pocket Outlook Object Model).
 OutlookSession **class**, 204–212, 218–224
 OutOfMemoryException, 290
 OverflowException **exception**, 23
over-the-air WAP push bootstrapping, 462

P

packaging applications, 56–59
 PaintEventArgs **control**, 387

painting, 387–388

Palm OS (Palm Inc.), 6

Panel control, 84–85

parameter passing, C# classes, 30

parametric polymorphism, 21

partial classes, new features, 438

passing

arrays, 266–267

blittable fields, 268–270

classes, 268–270, 270–271

multiple values, C# classes, 30

non-blittable fields, 270–271

parameters, C# classes, 30

by reference, 265–266

string variables, 267–268

structures, 268–270, 270–271

passwords, 330, 341–346

Path class, 112

Pen objects, 387–388

Pending() method, 179–187

performance

counters, 412–417

exceptions counters, 416

forms loading, 103

garbage collection counters, 416

general principles, 411–412

generics counters, 416

GUI-related counters, 417

interop counters, 416

JIT compilation counters, 416

loader counters, 416

memory usage counters, 416

.Net Compact Framework counters,
412–417

networking counters, 416

P/Invoke, 278

threads counters, 416

performance, CLR

BCL collections, 424–425

blittable data types, 419

call overhead, 418–423

class libraries, 424–428

COM calls, 418–420

foreach() method, 425

garbage collection, 417–418

generics, 423–424

iterating through collections, 425

math operations, 423

P/Invoke calls, 418–420

polymorphism, 423–424

Prelink() method, 419–420

PrelinkAll() method, 419–420

reflection operations, 423

Regex class, 427–428

regular expressions, 427–428

String class, 426–427

string handling, 426–427

StringBuilder class, 426–427

virtual calls, 420–423

performance, forms

BackgroundWorker class, 430–431

BeginUpdate() method, 433–434

EndUpdate() method, 433–434

InitializeComponent() method,
431–433

layout, 431–433

loading, 430–431

prepopulating data, 430–431

repainting controls, 433–434

ResumeLayout() method, 431–433

SuspendLayout() method, 431–433

performance, XML data access

local data, 429

reading XML data, 428

remote data, 429

writing XML data, 428

XML schema, 428

XML serialization, 429

XmlReader class, 428

XMLSerializer class, 429

XmlWriter class, 428

persistent storage filesystems, 106–107

philosopher's dinner problem, 300

Phone Lock feature, 330

physical access protection, 330

pictures. See graphics.

PIM data, 216–220. See also POOM (Pocket Outlook Object Model).

`PimItemCollection` **class**, 219–220

PINs, 330

P/Invoke

calls, performance, 418–420
 Dumpbin.exe utility, 258–260
 environment variables, Windows, 258–260
 managed DLLs, 257–260
 name mangling, 258

P/Invoke, managed code

converting to unmanaged, 263–271. *See also* marshaling.
 exporting functions from, 262
 memory management, 256
 portability, 256
 reusability, 256
versus unmanaged, 255–257

P/Invoke, .Net Compact Framework

calling conventions, 260
 character coding, 261
 error handling, 262–263
 example application, 271–277
 exceptions, 261
 exporting DLL functions, 261–262
 extern keyword, 262
 GetLastError property, 263
 marshaling data
 arrays, passing, 266–267
 blittable fields, passing, 268–270
 blittable types, 264–266
 classes, passing, 268–270, 270–271
 data fields, physical layout, 269–270
 definition, 263
 MarshalAs attribute, 270–271
 non-blittable fields, passing, 270–271
 out parameter, 265–266
 passing by reference, 265–266
 ref parameter, 265–266
 reference types, 266–268
 string keyword, 267–268
 string variables, passing, 267–268
 StringBuilder class, 267–268
 structures, passing, 268–270, 270–271
 UnmanagedType enumeration, 270–271
 value types, 264–266

MissingMethodException, 263
 NotSupportedException, 263
 overview, 16–17
 performance optimization, 278
 SetLastError property, 263
 static keyword, 262
 unidirection, 261
 unmanaged code functions, 261–262
 Windows messages, 261

PKI (public key infrastructure), 309

Platform Invoke. *See* P/Invoke.

platforms. *See* operating systems.

Pocket PCs, operating system, 7–8

Policy namespace, 453

polygons, drawing, 393–397

polymorphism

C#, 32–33
 performance, 423–424

POOM (Pocket Outlook Object Model), 202–203

POP (Power-On-Password), 330

Ports namespace, 450

power management, 7

PPTP (Point-to-Point Tunneling Protocol), 351

`Prelink()` **method**, 419–420

`PrelinkAll()` **method**, 419–420

`Priority()` **method**, 297

private keys, 309

private modifier, 31

`PrivateFontCollection` **class**, 386

privileged applications, 310

Privileged Applications Policy, 323

process window, 293

`ProcessConnection()` **method**, 189–192

processes, listing, 293

program memory, 106

programs, writing. *See* application development.

project files, 51

projects, creating, 43–49

properties, viewing, 293. *See also specific properties.*

Properties window, 46
protected memory exception, 290
protected modifier, 31
ProtocolVersion property, 168
Proxy property, 168
public key infrastructure (PKI), 309
public keys, 309
public modifier, 31
public-key encryption, 309

Q

Queue class, 34
QVGA issues, 102

R

race conditions, 298–300
RAM-based filesystems, 106
RAPI Policy, 323
RapiConfig tool, 460–461
Read() method, 178–187, 236–240
reading. See also forms; user input; user interface.
files, 108–112
XML data. performance, 428
ReadXml() method, 241–242
Receive() method, 189–196, 198–199
ReceiveFrom() method, 198–199
Recipient class, 206–208
Record button, 9
rectangles, drawing, 393–397
ref keyword, 30
ref parameter, 265–266
reference type objects, 19
reference types, 24, 266–268
reflection
.Net Compact Framework, 21
performance, 423
System.Reflection namespace, 452
Reflection namespace, 452

Regex class, 427–428
Region class, 389–391
regions, 389–391
Registry keys, new features, 438
regular expressions, performance, 427–428
RegularExpressions namespace, 454
relational data. See also databases; SQL Server 2005 Mobile.
ADO.NET, 132–134
connected mode, 132–133
converting from XML, 240–242
data providers, 132
disconnected mode, 132, 133–134
remote API, bootstrapping, 460–461
Remote Performance Monitor, new features, 439
remoting classes, 14
Remove() method, 27
request method does not exist, 290
request method not implemented, 290
resolution issues, 102
Resources namespace, 452
Restrict() method, 219–220
ResumeLayout() method, 431–433
return code error versus exceptions, 292
role-based access control, 308
roles, security, 311–312
ROM configuration XML file, bootstrapping, 462–464
root elements, 231, 233
Rotor, 12
runtime, 13
runtime inspection, 21
Runtime.CompilerServices namespace, 452
Runtime.InteropServices namespace, 452–453

S

Save As command, 102
schedules. See PIM data.
schema. See XML schema.

scroll, automatic, 69–78**searching**

- arrays, 33–34
- binary searches, 33–34
- `BinarySearch()` method, 33–34
- MSN Search, 228
- strings, 27

Secure Sockets Layer (SSL), 347–350**secured communications, 308****security**

- antivirus measures, 331–332
- application development, 7
- application signing, 309
- biometrics, 330
- bootstrapping, 464
- Cabir.A virus, 307–308
- CAS (Code Access Security), 308
- certificate stores, 310
- certificates, 309
- classes, 14
- device perimeter, 330–332
- digital signatures, 309
- errors, 290
- hash codes, 309
- loss, 308
- message digests, 309
- mobile threats, 307–308
- models, 59–60, 312–314
- .Net Compact Framework limitations, 308
- normal applications, 310–311
- passwords, 330
- Phone Lock feature, 330
- physical access, 330
- PINs, 330
- PKI (public key infrastructure), 309
- POP (Power-On-Password), 330
- private keys, 309
- privileged applications, 310
- public keys, 309
- public-key encryption, 309
- role-based access control, 308
- roles, 311–312
- secured communications, 308
- security filesystems, 308

- SIM (subscriber identity module), 330
- Symbian operating system, 307–308
- System.Security namespace, 453
- System.Security.Cryptography.X509Certificates namespace, 453
- System.Security.Policy namespace, 453
- theft, 308
- trusted applications, 310–311
- unprivileged applications, 310
- unsigned applications, 309
- untrusted applications, 310–311
- user authentication, 308
- vulnerability factors, 308

security, communications

- IPSec (IP Security), 351
- Kerberos, 347
- L2TP (Layer 2 Tunneling Protocol), 351
- network authentication, 346–347
- NTLM (NT LAN Manager), 346–347
- PPTP (Point-to-Point Tunneling Protocol), 351
- SSL (Secure Sockets Layer), 347–350
- tunneling protocols, 351–352
- VPNs (Virtual Private Networks), 350–352
- web communication, 347–350
- web services
 - anonymous access, 353–354
 - authentication, Windows domain servers, 353–354
 - client side, 360–362
 - digest authentication, 353–354
 - IIS (Internet Information Services), 353–354
 - Integrated Windows authentication, 353–354
 - server side, 354–359
 - user authentication, 353–354
- WEP (Wired Equivalent Privacy), 353
- WiFi, 353

security, data

- encryption
 - asymmetric cryptography, 335
 - cryptography algorithms, 334–335
 - data, 334–341
 - databases, 341–346
 - example, 335–341

security, data (continued)

- hardware, 334
- hashing cryptography, 335
- symmetric cryptography, 334
- password protection, 341–346

security, policies

- Auto Run, 322
- CABS, 324
- Grant Manager, 322
- Grant User Authenticated, 322
- Message Authentication Retry Number, 323
- Privileged Applications, 323
- RAPI, 323
- sample code, 325–329
- settings, 311–312
- SI (Service Indication), 323
- signing applications, 59–60
- SL (Service Loading), 323
- Trusted WAP Proxy, 324
- Unsigned Applications, 324
- Unsigned Prompt, 324

security, Windows Mobile 5.0

- certificates
 - loading, 321
 - MakeCert.exe tool, 319
 - managing, 319–321
 - obtaining, 314–315
 - Pvk2pfx.exe tool, 319
 - removing, 321
 - signing, 315–319
 - signtool utility, 319
 - viewing, 321
 - XML provisioning file, 321
- common Smartphone configurations, 313–314
- Locked setting, 314
- one-tiered security, 312
- One-Tier-Prompt setting, 313
- security model, 312–314
- Security-Off setting, 313
- Third-Party-Signed setting, 314
- two-tiered security, 313
- Two-Tier-Prompt setting, 314

Security namespace, 453

Security.Cryptography.X509Certificates namespace, 453

- SecurityException, 290
- Security-Off setting, 313**
- Security.Policy namespace, 453**
- SelectNode() method, 240
- SelectSingleNode() method, 240
- Send() method
 - EmailAccount class, 208
 - sending e-mail, 208
 - SmsAccount class, 221–225
 - sockets, blocking mode, 198–199
 - TCP client sockets, 192–196
 - TCP server sockets, 189–192
- SendTo() method, 198–199
- serial ports, new features, 439**
- server environment, 136**
- SetLastError property, 263
- SetPixel() method, 405–408
- shallow copy, C#, 24**
- shapes, drawing, 393–397**
- shortcut keys, menu items, 78–79**
- ShowResults() method, 252
- ShutDownServer() method, 189–192
- SI (Service Indication) messages, 61**
- SI (Service Indication) Policy, 323**
- signing applications, 59–61**
- SIM (subscriber identity module), 330**
- Skip() method, 236–237
- SL (Service Loading) messages, 61**
- SL (Service Loading) Policy, 323**
- Sleep() method, 296
- Smartphone**
 - connecting to PCs. See ActiveSync.
 - definition, 3–4
 - phone controls, 8–10
 - phone illustration, 9
 - user interface, 8–10
- smartphones**
 - definition, 4
 - network access, checking, 167
 - software platforms, 5
- smartphones, mobile uses for**
 - commerce, 5
 - data service, 6

- enterprise, 5
- entertainment, 6
- SMS, 220–225**
- SmsMessage **class, 221–225**
- Snapshot publication, 144–145**
- SOAP standard, 228**
- Socket **class, 187–188**
- sockets**
 - Accept() method, 198–199
 - asynchronous methods, 198–199
 - BeginReceive() method, 198–199
 - BeginSend() method, 198–199
 - blocking mode, 198–199
 - Connect() method, 192–196, 198–199
 - EndReceive() method, 198–199
 - EndSend() method, 198–199
 - nonblocking mode, 198–199
 - overview, 187–188
 - ProcessConnection() method, 189–192
 - Receive() method, 189–196, 198–199
 - ReceiveFrom() method, 198–199
 - Send() method, 189–196, 198–199
 - SendTo() method, 198–199
 - ShutDownServer() method, 189–192
 - Socket class, 187–188
 - SSL (Secure Sockets Layer), 347–350
 - StartServer() method, 189–192
 - TCP client, 192–196
 - TCP server, 188–192
 - UDP, 196–198
- soft keys, 9, 86, 101–102**
- software development. See application development.**
- solutions, 45**
- Sort() method, 33–34
- SortedList **class, 34**
- spaghetti dinner problem, 300**
- Specialized namespace, 443–444**
- SQL Server 2005 Mobile**
 - applications
 - databases, creating, 157–158
 - DataGrid control, example, 148–151
 - SqlServerCe namespace, 157–163
 - tables, adding rows, 158–160
 - web synchronization, 161–163
 - architecture, 135–136
 - client environment, 136
 - definition, 134
 - development environment, 135, 136
 - IIS (Internet Information Services), 136
 - installing. *See also* relational data.
 - client environment, 137, 138
 - development environment, 137
 - server environment, 137
 - system requirements, 136–137
 - new features, 134–135, 439
 - server environment, 136
 - setting up
 - databases, creating, 138–142, 147–148
 - publications, creating, 142–145
 - subscriptions, creating, 148–151
 - tables, creating, 138–142
 - tools, 138
 - web synchronization, configuring, 145–147
 - System.Data.SqlClient namespace, 446–447
 - System.Data.SqlServerCe namespace, 447
 - System.Data.SqlTypes namespace, 447–448
- SqlClient namespace, 446–447**
- SqlServerCe namespace, 157–163, 447**
- SqlTypes namespace, 447–448**
- SSL (Secure Sockets Layer), 347–350**
- Stack class, 34**
- stacking controls, 47**
- StackTrace **property, 289**
- Start() method, 296
- StartServer() method, 189–192
- static classes, 107**
- static **keyword, 262**
- static methods, C#, 23**
- storage cards, deploying applications, 61**
- storage efficiency, 6**
- Stream **class, 108–110, 112**
- StreamReader **class, 111–112**
- StreamWriter **class, 111–112**
- String **class, 426–427**

string handling, performance, 426–427

string keyword, 267–268

string operations. See C#, strings.

string reference type, 26–27

string variables, 26–27, 267–268

StringBuilder class, 27, 267–268, 426–427

StringReader class, 112

strings

drawing, 397–398

localization, 381–382

StringWriter class, 113

struct data types, 24

structures

passing, 268–270, 270–271

unsupported, 113

submenus, 78–79

subscriber identity module (SIM), 330

SuspendLayout() method, 431–433

Symbian OS (Symbian), 6, 307–308

symmetric cryptography, 334

synchronous web access, 169–170

System namespace, 442–443

System.Collections namespace, 34, 443

System.Collections.Generic namespace, 443

System.Collections.ObjectModel namespace, 443

System.Collections.Specialized namespace, 443–444

System.ComponentModel namespace, 445

System.Configuration.Assemblies namespace, 445

System.Convert class, 23

System.Data namespace, 445–446

System.Data.Common namespace, 446

System.Data.SqlClient namespace, 446–447

System.Data.SqlServerCe namespace, 447

System.Data.SqlTypes namespace, 447–448

System.Diagnostics namespace, 448

System.Drawing namespace, 448

System.Drawing.Drawing2D namespace, 449

System.Drawing.Imaging namespace, 449

System.Drawing.Text namespace, 449

SystemException class, 288–289

System.Globalization namespace, 449

System.IO namespace

BinaryReader class, 110–112

BinaryWriter class, 110–112

byte-level I/O classes, 108–110

classes, summary of, 112–113

classes, unsupported, 113

description, 450

Directory class, 107–108, 112

DirectoryInfo class, 107–108, 112

file access, 108–110

File class, 107–108, 112

file manipulation classes, 107–108

FileInfo class, 107–108, 112

FileStream class, 108–110, 112

FileStreamInfo class, 112

higher-level I/O classes, 110–112

IOException class, 112

MemoryStream class, 108–110, 112

Path class, 112

static classes, 107

Stream class, 108–110, 112

StreamReader class, 111–112

StreamWriter class, 111–112

StringReader class, 112

StringWriter class, 113

structures, unsupported, 113

TextReader class, 110–112, 113

TextWriter class, 110–112, 113

System.IO.Ports namespace, 450

System.Messaging namespace, 450–451

System.Net namespace, 451

System.Net.Sockets namespace, 451–452

System.Object class, 18

System.Reflection namespace, 452

System.Resources namespace, 452

System.Runtime.CompilerServices namespace, 452

System.Runtime.InteropServices namespace, 452–453

System.Security namespace, 453

System.Security.Cryptography.X509Certificates namespace, 453
System.Security.Policy namespace, 453
System.Text namespace, 453
System.Text.RegularExpressions namespace, 454
System.Threading namespace, 454–455
System.Web.Services namespace, 455
System.Web.Services.Description namespace, 455
System.Windows.Forms namespace, 455–456
System.XML namespace, importing, 234–235

T

T9 input mode, 97–97
tab order, controls, 47, 66–69
TabIndex property, 66–69
tables
 adding rows, 158–160
 creating, 138–142
TabStop property, 66–69
tasks. See PIM data.
TCP servers and clients
 AcceptSocket () method, 178–187
 AcceptTcpClient () method, 178–187
 big-endian byte order, 177–178
 byte order conversion, 177–178
 IP endpoints, 176–177
 IPAddress class, 176–177
 IPEndPoint class, 176
 IPHostEntry class, 176–177
 little-endian byte order, 177–178
 Pending () method, 179–187
 Read () method, 178–187
 TCP client, 180–182
 TCP server, 182–187
 TcpClient class, 178–187
 TcpListener class, 178–187
 WorkerThreadProc () method, 179–187
 TcpClient class, 178–187
 TcpListener class, 178–187
templates, 44
testing applications, 52–56
text
 alignment, 398
 drawing, 397–398
 wrap, 398
text messaging. See SMS.
Text namespace, 449, 453
 TextReader class, 110–112, 113
Text.RegularExpressions namespace, 454
 TextWriter class, 110–112, 113
theft, 308
Third-Party-Signed setting, 314
threading classes, 14
Threading namespace, 454–455
threads
 counters, 416
 debugging. *See* multithreaded debugging.
 new features, 439
 System.Threading namespace, 454–455
throw statement, 285–286
time, localization, 377–379
time zones, localization, 377–379
Timeout property, 168
timeout value, 168
 ToLower () method, 27
tools. See development tools.
 ToString () method, 23, 27, 289
 ToUpper () method, 27
Transactional publication, 144–145
TreeView control, 84–85
trusted applications, 310–311
Trusted WAP Proxy Policy, 324
 try...catch statement, 282–283
tunneling protocols, 351–352
two-tiered security, 313
Two-Tier-Prompt setting, 314
 typeof operator, 26
type-safe callback functions, 30–31

U

UDP, 196–198

UI. See user interface.

UI classes, 14

unchecked operator, 26

Undo command, 102

unhandled exceptions, 282, 289–290

unified directory view, 107

unmanaged code

accessing, 16–17

converting to managed, 263–271. *See also* marshaling.

definition, 11

`UnmanagedType` enumeration, 270–271

unprivileged applications, 310

unsigned applications, 309

Unsigned Applications Policy, 324

Unsigned Prompt Policy, 324

untrusted applications, 310–311

usability. See also user interface.

phone controls, 8–10

user-friendly GUI, 6

user authentication, 308

user input. See also controls; forms; user interface.

AlphaABC mode, 97

AlphaCurrent mode, 97

AlphaT9 mode, 97

ambiguous word input, 96–97

default behavior, 97

Default mode, 97

input mode, 96–101

multi-tap function, 96–101

Numeric mode, 97

T9 mode, 96–101

user interface

See also controls

See also forms

See also usability

See also user input

application flow, 87–88

auto-save mode, 102

card view, 87–88

list view, 87–88

modal dialog, 91

modeless dialog, 91

multiple forms, 88–96

performance, 103

QVGA issues, 102

resolution issues, 102

UTC (Coordinated Universal Time), 377

V

validating XML, 233–234

value type objects, 19

value types, C#, 23–24

variables, displaying, 293

vector graphics, 389–397

vector methods, 393–397

virtual calls, performance, 420–423

`virtual` keyword, 32–33

Visual Basic, namespaces, 441–442

Visual Studio 2005. See also debugging, with Visual Studio 2005.

application development, 40–41

development tools, 17–18

Edit and Continue feature, 55

overview, 14

VPNs (Virtual Private Networks), 350–352

W

watch windows, 293

watched variables, 55

web

communication security, 347–350

deploying applications from, 61

synchronization, 145–147, 161–163

web application classes, 14

web proxy, 168

web references, adding to web services, 248–249

web requests, creating

asynchronous access, 170–171

`Create()` method, 169

example, 171–175

overview, 169

synchronous access, 169–170

web services. *See also* security, web services; **XML (eXtensible Markup Language).**

- adding a web reference, 248–249
- advantages of, 228
- Amazon Web Services, 228
- anonymous access, 353–354
- authentication, Windows domain servers, 353–354
- classes, 14
- client side, 360–362
- consuming, 249–252
- definition, 228
- digest authentication, 353–354
- eBay Developers Program, 228
- Google Web APIs, 228
- IIS (Internet Information Services), 353–354
- Integrated Windows authentication, 353–354
- list of, 228–229
- MSN Search, 228
- server side, 354–359
- ShowResults() method, 252
- SOAP standard, 228
- System.Web.Services namespace, 455
- System.Web.Services.Description namespace, 455
- user authentication, 353–354
- WSDL (Web Services Description Language), 228
- Yahoo! Developer Network, 229
- WebRequest **class**, 167–175
- WebResponse **class**, 167–175
- Web.Services namespace, 455**
- Web.Services.Description namespace, 455**
- WEP (Wired Equivalent Privacy), 353**
- Whidbey.** *See* Visual Studio 2005.
- WiFi, 353**
- windows, debugging, 293–294**
- Windows Forms control, new features, 439–440**
- Windows Mobile 5.0**
 - See also* .Net Compact Framework
 - See also* .Net Framework
 - See also* security, Windows Mobile 5.0

overview, 7–8

Smartphone SDK, 41

WindowsCE Forms control, new features, 439

Windows.Forms namespace, 455–456

WindowsMobile.PocketOutlook namespace, 203–205

WorkerThreadProc() method, 179–187

world-ready applications. *See* globalization; localization.

wrapping exceptions, 286

WriteSchema mode, 242

writing

files, 108–112

programs. *See* application development.

XML data, performance, 428

WSDL (Web Services Description Language), 228

X

XML (eXtensible Markup Language). *See also* performance, XML data access; web services.

attribute values, reading, 236–237

converting to relational data, 240–242

current node, moving to root node, 236–240

data types, 234

depth-first node reading, 236

DiffGram mode, 242

document elements, 231

dumping data to, 241–242

examples, 231–233, 242–248

format standards, 227, 229

FormatOutput() method, 238–240

formatting output, 238–240

IgnoreSchema mode, 242

inline schemas, 234

Load() method, 235–240

loading files, 235–240

LoadXml() method, 240

MoveToAttribute() method, 236–237

MoveToContent() method, 236–240

XML (continued)

- .Net Compact Framework support, 229
- new features, 439–440
- non-data elements, skipping, 236–237
- processing error exceptions, 290
- querying strings, 240
- Read() method, 236–240
- ReadXml() method, 241–242
- related classes
 - DataSet, 240–242
 - summary of, 235
 - System.XML namespace, importing, 234–235
 - XmlDocument, 235–240
 - XmlNodeReader, 235, 240–242
 - XmlNodeWriter, 235
 - XmlReader, 235
 - XmlTextReader, 235–240
 - XmlTextWriter, 235
 - XmlWriter, 235
- root elements, 231, 233
- schema, 233–234
- SelectNode() method, 240
- SelectSingleNode() method, 240
- serialization, performance, 429
- Skip() method, 236–237
- strings, loading, 240
- subnodes, skipping, 236–237
- syntax, 229–231
- System.Xml namespace, 457

- System.Xml.Schema namespace, 457
- System.Xml.Serialization namespace, 457–458
- System.Xml.XPath namespace, 457–458
- validating, 233–234
- WriteSchema mode, 242

XML namespace, importing, 234–235

XML schema

- IgnoreSchema mode, 242
- inline schemas, 234
- overview, 233–234
- performance, 428
- System.Xml.Schema namespace, 457
- WriteSchema mode, 242
- XmlDocument **class, 235–240**
- XmlException, **290**
- XmlNodeReader **class, 235, 240–242**
- XmlNodeWriter **class, 235**
- XmlReader **class, 235, 428**
- XMLSerializer **class, 429**
- XmlTextReader **class, 235–240**
- XmlTextWriter **class, 235**
- XmlWriter **class, 235, 428**

Y

Yahoo! Developer Network, 229

