

# Index

COPYRIGHTED MATERIAL



# Index

## A

**accelerator keys, for heads-down data entry, 146**

### **Access database**

- ad hoc projects using, 83
- images from OLE object fields in, 234
- OleDb data provider for, 17

### **AccessDataSource control, 273**

**action verbs, FormView control, 292**

**AddNew method, BindingSource object, 166–167**

**Admin connection, SQL Server, 416**

**ADODB objects, compared to ADO.NET 2.0 objects, 2, 3, 5–6, 14**

**ADODB.Command object, 3**

**ADODB.Connection object, 3**

### **ADO.NET 2.0**

- autogenerating Windows forms
  - features for, 36
  - modifying autogenerated code, 132–133
  - procedure for, 28–30, 32–35
- data access features, 2–5
- data objects, basic
  - creating with Odbc, 19
  - creating with OleDb, 16–18
  - creating with SqlClient, 6–13
  - data-specific enumerations for, 19–21
  - list of, 5–6

- migrating to, 1–2

- persistent form objects, 69–70

- runtime objects, 39–40

- typed DataSets, 21–28, 30–31

### **AdventureWorks sample database**

- compared to Northwind sample database, 133

- xml columns in, 525–528

**adXactBrowse isolation level, 14**

**adXactChaos isolation level, 14**

**adXactCursorStability isolation level, 14**

**adXactIsolated isolation level, 14**

**AdXactReadCommitted isolation level, 14**

**AdXactReadUncommitted isolation level, 14**

**adXactRepeatableRead isolation level, 14**

**AdXactSerializable isolation level, 14**

**adXactUnspecified isolation level, 14**

**AgentMail, SQL Server, 430**

**Aggregate template, 459**

**alias data types, 491**

**Analysis Manager, SQL Server, 409**

**AntiPatterns Web site, 86**

**App\_Assemblies folder, 272**

**App\_Browsers folder, 272**

**App\_Code folder, 272, 325**

## **App.config file**

- adding Web Reference to, 362
- configuring for SQL Server Express, 133
- connection strings in, 119, 138
- setting UpdateColumns in, 546
- when not to save connection string in, 335, 381

## **App\_Data folder, 272**

## **App\_GlobalResources folder, 272**

## **Application Architecture for .NET: Designing Applications and Services, 84, 85–86**

## **application blocks**

- best practices for
  - Data Access Application Block (DAAB), 90–94
  - DataAccessQuickStart client, 94–96
  - list of application blocks, 89
  - using application blocks, 90
- definition of, 85, 89
- list of, 89

## **Application\_Code folder, 363**

## **App\_LocalResources folder, 272**

## **App\_Themes folder, 272**

## **App\_WebReferences folder, 272**

## **architectural best practices, 84–86**

## **ASP.NET 2.0**

- compilation model, 269–271
- creating Web applications, 266–269
- data-related server controls, 272
- DataSource controls, 273–274
- special (reserved) folders in, 271–272
- Web services features, 359–360

## **assemblies for SQL Server projects**

- creating, 466–468
- definition of, 457, 458

## **attribute decorations, SQL Server projects, 458–459**

## **authentication, for Web services, 372–374, 376–377**

## **autoenlisting in implicit transaction, 217–219**

## **autogenerating Windows forms, 28–30, 32–35, 36, 132–133**

## **automating test-driven development, 112–116**

# **B**

## **“Bad Code? FxCop to the Rescue” (Robbins), 110**

## **BaseColumnNamespace property, DataTableReader object, 45**

## **BaseTableNamespace property, DataTableReader object, 45**

## **Basic credentials, 373**

## **basic data objects**

- creating with Odbc, 19
- creating with OleDb, 16–18
- creating with SqlClient, 6–13
- data-specific enumerations for, 19–21
- list of, 5–6

## **Basic Profile (BP), WS-I, 106**

## **batch inserts, 47–50**

## **batch queries, using instead of stored procedures, 122**

## **batch updates, 77–79, 126, 169**

## **BEGIN CATCH statement, T-SQL, 417–418**

## **BEGIN TRY statement, T-SQL, 417–418**

## **BeginExecuteNonQuery method, SqlCommand object, 52**

## **BeginExecuteReader method, SqlCommand object, 52**

## **BeginExecuteXmlReader method, SqlCommand object, 52**

## **BeginTransaction method, SqlTransaction object, 13**

## **BEs (business entities), 97–102**

## **best practices**

- application blocks
  - Data Access Application Block (DAAB), 90–94
  - DataAccessQuickStart client, 94–96
  - list of, 89
  - using, 90
- architectural, 84–86
- automating test-driven development, 112–116
- batch queries, using instead of stored procedures, 122
- batch updates, 126
- class library design guidelines for, 102–103
- client-side validation, 126–127

- CommandBuilder object, not instantiating at runtime, 121
- concurrency tests, checking all related records for, 126
- default values for parameters, 122
- definition of, 84
- design guides for, 96–102
- FxCop code analysis tool, 110–111
- for heads-down data-entry forms, 146–147
- inspecting queries with SQL Server Profiler, 120–121
- for offline DataSet updates, 209
- optimistic concurrency control, 123–125
- patterns, 86–89
- pooling database connections, 118–120
- reusing cached query plans, 122–123
- service-oriented architecture (SOA)
  - definition of, 103
  - ensuring interoperability, 106–110
  - history of, 104
  - implementing with Web services, 105–106
- SQL Server 2000 Best Practices Analyzer (BPA), 116–118
- Best Practices Analyzer (BPA), 116–118**
- bidirectional cursors, 6**
- BIDS (Business Intelligence Development Studio), 410**
- BindingList collection, 259–261**
- BindingNavigator control**
  - adding to form, 28–30
  - definition of, 29, 70
- BindingSource object**
  - adding records to, 149, 150, 166–167
  - definition of, 70
  - synchronization of DataGridViews by, 152
- bound controls**
  - DataBoundControl class, 274
  - DataList control
    - definition of, 274
    - editing items in, 286–288
    - formatting `datetime` values, 281–282
    - formatting `money` values, 281–282
    - properties of, 280
    - SqlDataSource for, 275–279
    - templates for, 281–283
  - DetailsView control
    - adding ObjectDataSource control to, 332–333
    - assigning default values, 304
    - definition of, 303
    - displaying business objects, 339–343
    - editable composite primary keys, 304
    - errors in updates or insertions, 304–305
    - linking to GridView page, 306–307
    - SqlDataSource for, 275–279
    - synchronizing with child table GridView, 304
    - validating user input in, 310, 322–324
  - formatting `datetime` values, 281–282
  - formatting `money` values, 281–282
  - FormView control
    - command buttons for action verbs, 292–293
    - definition of, 288
    - editing, adding, deleting records in, 291–292
    - paging, 288–289
    - replacing null values, 289–290
    - SqlDataSource for, 275–279
  - GridView control
    - adding ObjectDataSource control to, 328–332
    - definition of, 293–295
    - displaying business objects, 339–343
    - field types in, 295
    - ImageFields in, 299–303
    - linking to DetailsView page, 306–307
    - populating with XML file, 346
    - replacing TextBoxes with DropDownList, 297–298
    - SqlDataSource for, 275–279
    - synchronizing with DetailsView, 306–307
    - validating user input in, 295–297, 313–321
    - validation groups and, 311
    - list of, 274
    - properties of, 280
  - Repeater control
    - design with XmlDataSource control, 347–348
    - SqlDataSource for, 275–279
  - SqlDataSource for, 275–279
  - templates for, 281–283
  - TreeView control
    - populating from XmlDataSource control, 348–350
    - SqlDataSource for, 275–279
  - WHERE clause constraints, 283–286

### **BoundFields, GridView control**

converting to TemplateFields, 295–297  
definition of, 295

### **BP (Basic Profile), WS-I, 106**

### **BPA (Best Practices Analyzer), 116–118**

### **BPCs (business process components), 97**

### **brittleness of applications, 104**

### **business entities (BEs), 97–102**

### **Business Intelligence Development Studio (BIDS), 410**

### **business objects, custom**

creating ObjectDataSource control from, 336–343

with Web services

as alternative to DataSets, 384–391

binding to DataGridViews, 391–395

client for, 397–399

updating, inserting, deleting, 395–397

using, issues regarding, 361

### **business process components (BPCs), 97**

### **business rules, 168–169**

### **ButtonFields, GridView control, 295**

## **C**

### **cached query plans, reusing, 122–123**

### **callback model, asynchronous SqlCommand execution, 55–58**

### **camelCase, 102**

### **CancelEdit method, BindingSource object, 166–167**

### **CancelNew method, BindingSource object, 166–167**

### **cascading updates, 170**

### **case, for class and member names, 102–103**

### **Cassini Web server, 265**

### **CATCH blocks, T-SQL, 417–418**

### **CellValidated event, 184**

### **CellValidating event, 184**

### **CFS (CLR scalar user-defined function)**

creating, 468–469

definition of, 458, 459

example of, 462

### **CFT (CLR table-valued user-defined function)**

creating, 469

definition of, 458, 459

### **Chaos isolation level, 14**

### **CheckBoxFields, GridView control, 295**

### **CheckConstraints member, SqlBulk**

**CopyOptions** enumeration, 48

### **child rows, testing for concurrency errors, 194–195**

### **child-record count mismatches, detecting, 192–194**

### **class library design guidelines, 102–103**

### **classes, serializable**

creating data sources from, 257–259

generating from XML schema, 255–261

### **clients of Web services**

DataGridView control for, 380–382

proxies for, 369–372

### **client-server architecture, 104, 131**

### **client-side validation**

best practices for, 126–127

enabling, 311

example of, 323–324

### **CLR integration, SQL Server**

definition of, 413

enabling, 458

SQL/CLR objects

attribute decorations identifying, 458–459

creating, 468–470

definition of, 457

dropping, 470

examples of, 461–465

performance and, 458

when to use, 457

### **CLR scalar user-defined function (CFS)**

creating, 468–469

definition of, 458, 459

example of, 462

### **CLR stored procedure (CSP)**

creating, 468

definition of, 458, 459

examples of

generating XML documents, 475–484

projecting product sales, 484–491

returning content-dependent

SqlDataRecords, 473–475

returning specified order rows, 461–462

when to use, 472

- CLR table-valued user-defined function (CFT)**
  - creating, 469
  - definition of, 458, 459
- CLR trigger (CTR)**
  - creating, 469–470
  - definition of, 458, 459
  - example of, 464–465
- code analysis tool, FxCop, 110–111**
- coding, best practices for. See best practices**
- coincidence, leading to pattern, 86**
- ColumnMapping property,**
  - DataTableReader object, 45**
- ColumnSize property**
  - DataReader object, 45
  - DataTableReader object, 45
  - SqlDataReader object, 45
- Command object, 5, 6**
- CommandBehavior.CloseConnection flag, 11**
- CommandBehavior.SingleRow flag, 11**
- CommandBuilder object**
  - not instantiating at runtime, 109, 121
  - using for updates, 382–383
  - using in transactions, 214–217
- CommandFields, GridView control, 295**
- CommittableTransaction object, 217, 220–221**
- CommitTransaction method,**
  - SqlTransaction object, 13, 14**
- Common namespace, 3**
- common table expressions (CTEs), T-SQL, 421–422**
- CompareValidator control, 311, 318**
- compensating transactions, 360**
- ComponentModel namespace, 3, 259**
- composite primary keys, 334**
- concurrency control, optimistic, 123–125**
- concurrency management**
  - approaches to, 189
  - child rows, testing for concurrency errors, 194–195
  - child-record count mismatches, detecting, 192–194
  - by DataAdapters, 190–191
  - definition of, 181, 189
  - deleted records, allowing recreation of, 195–198
  - handling errors
    - comparing client and server values, 203–208
    - getting current data from server for, 202–203
    - methods of, 201–202
    - new features for, 189–191
    - primary key constraint violations, catching, 198–201
    - specification for, 191–192
    - by TableAdapters, 190–191
    - transactions for, 190
- concurrency tests, checking all related records for, 126**
- configuration files**
  - App.config file
    - adding Web Reference to, 362
    - configuring for SQL Server Express, 133
    - connection strings in, 95, 119, 138
    - setting UpdateColumns in, 546
    - when not to save connection string in, 335, 381
  - Web.config file
    - adding to project, 268
    - adding Web Reference to, 362
    - ConfigurationManager class for, 379
    - connection strings in, 119, 120, 325, 376
    - enabling anonymous Web service users, 377
    - removing debugging symbols from temporary files, 355, 369
    - specifying multiple traces in, 352
- Configuration Manager, SQL Server, 407–408**
- Connection control, adding to toolbox, 72**
- Connection object, 5, 6**
- connection statistics, SQL Server, 51–52**
- connection strings**
  - in configuration files
    - encryption of, 120
    - how to store, 119, 325, 376
    - modifying, 138
  - encrypting, 120
  - for Web services, 376–377
  - when not to save in configuration files, 335, 381

### **connections, database**

- connection statistics, SQL Server, 51–52
- connection strings
  - in configuration files, 119, 120, 138, 325, 376
  - encrypting, 120
  - for Web services, 376–377
  - when not to save in configuration files, 335, 381
- pooling, best practices for, 118–120
- storing connection strings in configuration files, 119
- constraint violations for primary key, catching, 186–187, 198–201**
- ContinueUpdateOnError property**
  - DataAdapter object, 191
  - TableAdapter object, 191
- controls. See also specific controls**
  - data-related server controls, 272
  - DataSource controls, 273–274
  - non-standard, adding to toolbox, 72
  - Web controls
    - for data binding and updates, list of, 272, 273
    - for data validation, list of, 310–311
- ControlToValidate property, 311**
- CREATE ASSEMBLY instruction, 466–468**
- CREATE ObjectType instructions, 468–470**
- CreateCommand method,**
  - DbProviderFactories object, 17
- CreateConnection method,**
  - DbProviderFactories object, 17
- credentials. See also connection strings**
  - Connection object for, 5
  - for Web service authentication, 372–374, 377
- Credentials property,**
  - SoapHttpClientProtocol class, 371
- CRM (customer relationship management), 235**
- cross-domain queries, 513**
- CSP (CLR stored procedure)**
  - creating, 468
  - definition of, 458, 459
  - examples of
    - generating XML documents, 475–484
    - projecting product sales, 484–491

- returning content-dependent
  - SqlDataRecords, 473–475
- returning specified order rows, 461–462
- when to use, 472

### **CTEs (common table expressions), T-SQL, 421–422**

#### **CTR (CLR trigger)**

- creating, 469–470
- definition of, 458, 459
- example of, 464–465

#### **currency values, formatting, 281–282**

#### **custom business objects**

- creating ObjectDataSource control from, 336–343
- with Web services
  - as alternative to DataSets, 384–391
  - binding to DataGridViews, 391–395
  - client for, 397–399
  - updating, inserting, deleting, 395–397
  - using, issues regarding, 361

#### **customer orders example**

- basic data entry form for, 132–138
- business rules, applying, 168–169
- combo boxes in
  - associating with text boxes, 159–161
  - setting additional values with, 162–165
  - text boxes replaced with, 156–159
- data source for, 133–134
- DataGridView controls
  - adding computed column to, 141–143
  - creating, 133–134
  - formatting, 140–141
  - handling DataErrors event for, 146
  - populating, 137
- DataTables, populating, 135–136
- default values for new records, 143–145, 150–151
- heads-down data entry, streamlining, 146–148
- lookup lists for primary key values, 153–161
- new customers, adding lookup table rows for, 166–169
- referential integrity, maintaining, 170
- saving changes to base tables
  - maintaining referential integrity, 170
- procedure for, 171–180
- strategy for, 169
- tabbed form, migrating data entry form to, 149–153

**customer relationship management (CRM),**  
235

**CustomValidator control, 311, 319–320**

## D

**DAAB (Data Access Application Block), 90–94**

**DALCs (data access logic components)**

creating ObjectDataSource control from,  
335–336

definition of, 97

ObjectDataSource control enabling, 325

**Data Access Application Block (DAAB), 90–94**

**data access features, 2–5**

**data access logic components (DALCs)**

creating ObjectDataSource control from,  
335–336

definition of, 97

ObjectDataSource control enabling, 325

**data binding. See DataSet object**

**Data Definition Language (DDL) triggers, 424**

**data encryption, SQL Server, 424–425**

**data entry**

autogenerated forms for, 36, 132–133

creating, process for, 70–71

database-agnostic applications for, 40

heads-down, 146–148

**Data Mining Extensions (DMX) scripts, 409**

**Data namespace, 2–4**

**data objects, basic**

creating with Odbc, 19

creating with OleDb, 16–18

creating with SqlClient, 6–13

data-specific enumerations for, 19–21

list of, 5–6

**Data Patterns, 84, 87–88**

**data providers. See also specific data providers**

all available, generating connection and  
commands for, 17

checking for available providers, 46–47

by DataDirect Technologies, 40–41

list of, 4–5

namespaces for, 3, 4

**data source**

creating from serializable classes, 257–259

definition of, 22

generating typed DataSet object from, 24–28

**data source-agnostic projects, 40–43**

**Data Sources window**

adding data source, 24, 71, 73

adding related table to, 32, 35, 73

creating DataGridViews from, 28

creating Details from, 31

multiple tables in, 73–75

**Data Transfer Object (DTO), 87**

**data types**

alias data types, 491

namespace for, SQL Server, 4

nullable types, 40, 66–69

UDDTs (user-defined data types), 491–492

xml data type, SQL Server

definition of, 413–414, 513–514

indexed xml columns, 522–525

sample database containing, 525–528

typed xml columns, 519–522

untyped xml columns, 515–518

**data validation**

client-side validation, 126–127, 311,  
323–324

for DataGridView control, 184–186

for default values, 187–188

definition of, 181, 183, 310

for GridView control

adding controls to template, 313–315

CompareValidator control for, 318

CustomValidator control for, 319–320

RangeValidator control for, 316–318

RegularExpressionValidator control for,  
315–316, 317–318

ValidationSummary control for, 321

primary key constraint violations, catching,  
186–187

for text boxes, 183–184

validation groups, 311

Web controls for

list of, 310–311

properties of, 311–312

**DataAccessQuickStart client, 94–96**

**DataAdapter control**

adding to toolbox, 72

batch updates using, 169

concurrency management by, 190

updates using, 172

### **database connections**

- connection statistics, SQL Server, 51–52
- connection strings
  - in configuration files, 95, 119, 138
  - encrypting, 120
  - for Web services, 376–377
  - when not to save in configuration files, 335, 381
- pooling, best practices for, 118–120
- storing connection strings in configuration files, 119

### **Database Mail, SQL Server, 430**

### **database mirroring, SQL Server, 416**

### **Database Tuning Advisor, SQL Server, 412**

### **databases, sample**

- AdventureWorks sample database
  - compared to Northwind sample database, 133
  - xml columns in, 525–528
- Northwind sample database
  - compared to AdventureWorks sample database, 133
  - configuration change for Access MSDE 2000 server, 7
  - configuration change for SQL Server Express, 39, 133
  - size of, 132

### **DataBoundControl class, 274**

### **DataDirect Technologies, data providers by, 40–41**

### **DataErrors event, DataGridview control, 146**

### **DataGridview control. See also GridView control**

- adding to form, 133–134
- combo boxes in, associating with text boxes, 159–161
- computed columns in, 141–143
- DataErrors event for, handling, 146
- default values for new records, 143–145
- definition of, 70
- error provider for, 181, 184
- formatting columns in, 140–141
- images in, 229–234
- joins in, 224–227
- lookup lists for primary key values, 153–161

- populating
  - with DataReader object, 10–11
  - with DataSet object, 28–30, 32–35
- text boxes in, replacing with combo boxes, 156–159
- validating, 184–186
- for Web service client, 380–382

### **DataList control**

- definition of, 274
- editing items in, 286–288
- formatting datetime values, 281–282
- formatting money values, 281–282
- properties of, 280
- SqlDataSource for, 275–279
- templates for, 281–283

### **DataNavigator control, 28–29, 31**

### **DataReader object**

- creating from DbProviderFactories class, 41
- definition of, 5, 6
- inserting rows from, to base table, 48
- populating DataGridView control with, 10–11
- properties of, 44–45
- table schemas and, 43–46
- typed data from, 19–21

### **data-related server controls in ASP.NET 2.0, 272**

### **DataSet object**

- adding to existing Web site, 326–327
- compared to disconnected Recordsets, 21–22
- creating from SQL Server data source, 22–28
- creating in ADO.NET 1.x, 70–71
- creating in ADO.NET 2.0, 71
- creating ObjectDataSource control from, 325–334
- custom business objects as alternative to, 384–391
- definition of, 21
- files generated by, 26–27
- generating from adapted XML schema
  - considerations for, 235–236
  - flat schemas, 245–248
  - nested schemas, 236–245
- locally persisted DataSets, 209
- lookup DataSets, 154–155

persisting and reopening, 30–31  
 populating DataGridView control with, 28–30, 32–35  
 populating details form with, 31–32  
 returning in Web service, 377–380  
 size of, reducing, 132–133, 227–229  
 updating base tables from  
   maintaining referential integrity, 170  
   procedure for, 171–180  
   strategy for, 169  
 updating in a transaction, 214–221  
 updating in Web service, 382–384  
 updating offline DataSets, 208–212  
 with Web services, 361

**DataSetWS Web service**

installing and publishing, 107–109  
 testing, 109–110  
 VSTS suite used with, 112–116

**DataSource controls**

definition of, 273–274  
 list of, 273  
 performance of, 351–353

**DataSourceControl class, 274****DataSourceView object, 274****DataTable object**

adding to existing Web site, 326–327  
 batch updates to, 77–79  
 creating ObjectDataSource control from, 325–334  
 features for, 40  
 lookup lists for, 154–156  
 populating, 135–136, 137  
 snapshot data stored by, 189  
 standalone, 62–66  
 table schemas and, 43–46

**DataTableAdapter object**

concurrency management for, 181  
 limiting DataSet size with, 227  
 reconfiguring to use stored procedures, 71

**DataTableReader object**

properties of, 44–45  
 table schemas and, 43–46

**DataType property**

DataReader object, 45  
 DataTableReader object, 45  
 SqlDataReader object, 45

**DataGridView object**

retrieving current data from, 202–203  
 sorted, creating and binding, 162–163  
 sorting performance of, 194

**dates, validating, 317–318****datetime values, formatting, 281–282****DateTimePicker control, problems with, 152****DBConcurrencyException exception**

definition of, 181  
 methods of handling, 201–202

**DbConnection namespace, 3****DBNull value, 40, 66–69****DbProviderFactories class, 17, 40–43****DbProviderFactory object**

creating, 41  
 definition of, 40

**DbType enumeration, 20****DDL (Data Definition Language) triggers, 424****debugging SQL Server projects, 471–472****Default member, SqlBulkCopyOptions enumeration, 48****default values**

best practices for, 122  
 data validation for, 187–188  
 setting for new records, 143–145, 150–151

**Default.aspx file, 267****dehydrating an object, 256****deleted records, allowing recreation of, 195–198****DENSE\_RANK() function, T-SQL, 420–421****deserializing an object, 256****“Design Guidelines for Class Library Developers”, 102–103****design guides, 85, 96–102****Designing Data Tier Components and Passing Data Through Tiers, 85, 97–102****details form, 31–32****DetailsView control**

adding ObjectDataSource control to, 332–333  
 assigning default values, 304  
 definition of, 303  
 displaying business objects, 339–343  
 editable composite primary keys, 304  
 errors in updates or insertions, 304–305  
 linking to GridView page, 306–307  
 SqlDataSource for, 275–279

### DetailsView control (continued)

synchronizing with child table GridView, 304  
validating user input in  
  client-side validation, 323–324  
  EditItemTemplate fields for, 310, 322  
  InsertItemTemplate fields for, 322  
  server-side validation, 322–323

### Developer Edition, SQL Server, 405

development, best practices for. *See* best practices

### Digest credentials, 373

disconnected client scenario, 86

disconnected users, enabling data modifications by, 182, 208–212

DisplayMode property, 311

*Distributed Systems Patterns*, 88

DMO (SQL-DMO, SQL Distributed Management Objects), 411

DMX (Data Mining Extensions) scripts, 409

drag-and-drop methods, 36, 39–40

DTO (Data Transfer Object), 87

## E

EAI (enterprise application integration) patterns, 88

editing forms, creating from XML documents, 250–254

EditItemTemplate control, in GridView control, 295–297, 310, 313–315

EditMode property, DataGridView control, 147

Electronic Commerce Dictionary, 88

EnableClientScript property, 311

EnableDecompression property, SoapHttpClientProtocol class, 371, 374

EnableStatistics property, SqlConnection object, 51

### encryption

  connection strings, 120  
  SQL Server, 424–425

END CATCH statement, T-SQL, 417–418

END TRY statement, T-SQL, 417–418

EndEdit method, BindingSource object, 166–167

EndExecuteNonQuery method, SqlCommand object, 52

EndExecuteReader method, SqlCommand object, 52

EndExecuteXmlReader method, SqlCommand object, 52

enlistment of resource manager for transactions, 217–221

enterprise application integration (EAI) patterns, 88

Enterprise Development Reference Architecture (ERDA), 86

Enterprise Edition, SQL Server, 406

Enterprise Library application blocks, 89–90

Enterprise Manager, SQL Server, 409

*Enterprise Solution Patterns Using Microsoft .NET*, 87

ERDA (Enterprise Development Reference Architecture), 86

error icon, 181

ErrorMessage property, 311

ErrorProvider object, 181, 183

errors, concurrency

  current data from server, getting, 202–203  
  methods of handling, 201–202  
  resolving by comparing client and server values, 203–208

### events

  for data validation, 183–184  
  DataErrors event, handling, 146  
  leading to patterns, 86

eWeek article on Gartner market share numbers, 403

### examples

  AdventureWorks sample database compared to Northwind sample database, 133

  xml columns in, 525–528

  customer orders example

    basic data entry form for, 132–138

    business rules, applying, 168–169

    combo boxes in, 156–165

    data source for, 133–134

    DataGridView controls, 133–134, 137, 140–143, 146

- DataTables, populating, 135–136
- default values for new records, 143–145, 150–151
- heads-down data entry, streamlining, 146–148
- lookup lists for primary key values, 153–161
- new customers, adding lookup table rows for, 166–169
- referential integrity, maintaining, 170
- saving changes to base tables, 169–180
- tabbed form, migrating data entry form to, 149–153
- Northwind sample database
  - compared to AdventureWorks sample database, 133
  - configuration change for Access MSDE 2000 server, 7
  - configuration change for SQL Server Express, 39, 133
  - size of, 132
- exclamation mark icon, 181**
- ExecuteNonQuery method, SqlCommand object, 12–13, 14**
- ExecutePageReader method, 6**
- ExecuteReader method, SqlCommand object, 7**
- ExecuteScalar method, SqlCommand object, 14**
- ExecuteXmlReader method, SqlCommand object, 8–9**
- Express Edition, SQL Server, 404–405**
- Expression property, DataTableReader object, 45**
- F**
- fields, usage guidelines for, 103**
- FillBy methods, 132, 135–136**
- FireTriggers member, SqlCommandBulkCopyOptions enumeration, 48**
- flat XML schema, 245–248**
- FLWOR XQuery expressions, 535–539**
- folders**
  - special (reserved) folders in ASP.NET 2.0, 271–272
  - for Web services, 363
- FOR XML AUTO clause, T-SQL, 8–10, 422–424, 432**
- FOR XML EXPLICIT clause, T-SQL, 432**
- FOR XML PATH clause, T-SQL, 435–440**
- FOR XML RAW clause, T-SQL, 432–435**
- foreign-key constraints, 170**
- forms (Web). See also bound controls**
  - compilation model, 269–271
  - creating, process for, 266–269
  - scalability requirements, 2
  - Web controls
    - for data binding and updates, list of, 272, 273
    - for data validation, list of, 310–311
- forms (Windows). See also DataGridView control**
  - autogenerating
    - features for, 36
    - modifying autogenerated code, 132–133
    - procedure for, 28–30, 32–35
  - data entry forms
    - autogenerating, 36, 132–133
    - creating, process for, 70–71
    - database-agnostic applications for, 40
    - heads-down, 146–148
  - details form, 31–32
  - drag-and-drop methods for, 36, 39–40
  - editing forms, creating from XML documents, 250–254
  - master-detail forms, 71
  - master-detail, parameterized, 75–77
  - multi-level subforms, 73–75
  - objects for, 39–40
  - persistent form objects
    - adding ADO.NET 1.x components to toolbox, 72
    - batching multiple updates, 77–79
    - comparing ADO.NET 1.x to 2.0, 70–71
    - list of, 69–70
    - multi-level subforms, 74–75
    - parameterized MasterDetailsForm, 75–77
    - upgrading to ADO.NET 2.0, 72–73
  - tabbed form, migrating data entry form to, 149–153

### **FormView control**

- command buttons for action verbs, 292–293
- definition of, 288
- editing, adding, deleting records in, 291–292
- paging, 288–289
- replacing null values, 289–290
- SqlDataSource for, 275–279

**FxCop code analysis tool, 110–111**

## **G**

**Gartner market share numbers for SQL Server, 403**

**GBI (Global Bank Implementation) project, 86**

**Generics namespace, 259**

**GetDataSources method, 46–47**

**GetFactory method,**

- DbProviderFactories object, 17**

**GetFactoryClasses method,**

- DbProviderFactories class, 46–47**

**GetSchemaTable method**

- DataReader object, 43–46

- DataTableReader object, 43–46

**Global Bank Implementation (GBI) project, 86, 88**

**GridView control. See also DataGridView control**

- adding ObjectDataSource control to, 328–332
- definition of, 293–295
- displaying business objects, 339–343
- field types in, 295
- ImageFields in, 299–303
- linking to DetailsView page, 306–307
- populating with XML file, 346
- replacing TextBoxes with DropDownLists, 297–298
- SqlDataSource for, 275–279
- synchronizing with DetailsView, 306–307
- validating user input in
  - adding controls to template, 313–315
  - CompareValidator control for, 318
  - CustomValidator control for, 319–320
  - EditItemTemplate fields required for, 295–297, 310, 313–315
  - RangeValidator control for, 316–318

- RegularExpressionValidator control for, 315–316, 317–318

- ValidationSummary control for, 321

- validation groups and, 311

**Gzip compression, 361**

## **H**

**heads-down data entry, 146–148**

**help page, for Web services, 364–369**

**HR-XML (Human Resources XML), 236**

**HTTPS protocol, 363**

**hydrating an object, 256**

**HyperlinkFields, GridView control, 295**

## **I**

**IDataReader object, 41**

**IDataSource interface, 274**

**IDbCommand object, 41**

**IDbConnection object, 41**

**IIS virtual directory, deploying Web applications to, 353–356**

**Image Web server control, 302–303**

**ImageFields, GridView control**

- creating, 299–301

- definition of, 295

- scaling, 302–303

**images**

- from Access OLE object fields, 234

- in DataGridView control

  - image columns for, 230–231

  - layout for, 231–232

- saving and displaying from a file, 232–234

- scaling, 302–303

**“Improving .NET Application Performance and Scalability”, 96–97**

**indexed xml columns**

- add and drop XML indexes, 523–524

- definition of, 522–523

- performance of, 546–557

- promoting xml values to table columns, 524–525

**InsertItemTemplate control, in DetailsView control, 322**

**inserts**

- batch inserts, 47–50
- errors in, 304–305

**Integration Patterns, 88–89****Integration Services, SQL Server, 411****IsInRole method, 147****IsLong property**

- DataReader object, 45
- DataTableReader object, 45
- SqlDataReader object, 45

**isolation levels**

- SqlTransaction object, 13–14
- T-SQL, 418–419

**IsolationLevel property,**

- SqlTransaction object, 13, 14

**IsolationLevelEnum property, 14****J****Jet databases, ad hoc projects using, 83****joins**

- adding to SelectCommand object, 222–224, 225–226
- in DataGridView control, 224–227

**K****KeepIdentity member,**

- SqlBulkCopyOptions enumeration, 48

**KeepNulls member, SqlBulkCopyOptions enumeration, 48****L****Language Integrated Query (LINQ) project, 261****“last user wins” approach, 189****latency of snapshot data, 189, 209****Lightweight Transaction Manager (LTM), 217, 218****linear regression analysis, projecting product sales using, 484–491****LINQ (Language Integrated Query) project, 261****ListControl class, 274****long-running transactions, 360****lookup lists for primary key values, 153–161****LTM (Lightweight Transaction Manager), 217, 218****M****Management Studio, SQL Server, 409–410****MARS (Multiple Active Result Sets), 415–416****MarshalByRefObject namespace, 3****master-detail forms**

- definition of, 71
- parameterized, 75–77

**MDX (Multidimensional Expressions) scripts, 409****member usage guidelines for class libraries, 103****methods, usage guidelines for, 103. See also specific methods****Microsoft Architects JOURNAL, 85****Mobile Edition, SQL Server, 406****mode verbs, FormView control, 292****modify method, 513****money values, formatting, 281–282****monolithic architecture, 104****MSDN Web Services Developer Center, 106****Multidimensional Expressions (MDX) scripts, 409****“Multiple Active Result Set (MARS) in SQL Server 2005” (white paper), 416****Multiple Active Result Sets (MARS), 415–416****N****named pipes (Np), SQL Server, 408****naming guidelines for class libraries, 102–103****native SOAP Web services, SQL Server, 430–432, 453–455****navigable cursors, 6****nested XML schema, generating DataSet from, 236–245****.NET Data Access Architecture Guide, 85, 96****NextResult method, SqlDataReader object, 7****Northwind sample database**

- compared to AdventureWorks sample database, 133
- configuration change for Access MSDE 2000 server, 7
- configuration change for SQL Server Express, 39, 133
- size of, 132

**Notification Services, SQL Server, 409, 429**

**Np (named pipes), SQL Server, 408**

**n-tier architecture, 104**

**nullable types, 40, 66–69**

## O

**Object namespace, 3**

**ObjectDataSource control**

adding to DetailsView control, 332–333

adding to GridView control, 328–332

compared to SqlDataSource control, 325

creating from custom business objects,  
336–343

creating from DALCs, 335–336

creating from DataTable objects, 325–334

definition of, 273

enabling DALCs, 325

**object-relational mapping (OR-M), 387**

**objects. See also specific objects**

ADODB objects, 2, 3, 5–6, 14

data objects, basic, 5–6

persistent form objects

adding ADO.NET 1.x components to  
toolbox, 72

batching multiple updates, 77–79

comparing ADO.NET 1.x to 2.0, 70–71

list of, 69–70

multi-level subforms, 74–75

parameterized MasterDetailsForm, 75–77

upgrading to ADO.NET 2.0, 72–73

re-hydrating an object, 256

runtime objects, 39–40

**ObjectSpaces framework, 387**

**Odbc data provider, 19**

**odbc namespace, 4, 5**

**offline DataSets, 208–212**

**OLE object fields, images from, 234**

**OleDb data provider**

creating basic data objects, 16–18

returning XmlReader objects, 18

**OleDb namespace, 4, 5**

**OleDbCommand object, 17–18**

**optimistic concurrency control, 189**

**OracleClient namespace, 4**

**OR-M (object-relational mapping), 387**

**OTM (OleTx Transaction Manager), 217**

## P

**PAG (Platform Architectural Guidance)  
team, 84**

**paging, FormView control, 288–289**

**parameterized queries**

creating, 75–77

default values for, 122

reducing DataSet size with, 132–133

reusing cached query plans with, 122–123

**partial classes for TableAdapter object,  
228–229**

**partitioning, SQL Server, 416**

**PascalCase, 102**

**passwords**

authentication, for Web services, 372–374,  
376–377

credentials

Connection object for, 5

for Web service authentication, 372–374,  
377

**PATH XML index, 522**

**pattern template, 87**

**patterns**

best practices for, 86–89

definition of, 84

**patterns and practices (P&P), by Microsoft,  
84–86. See also best practices**

**performance**

batch queries and, 122

batch updates and, 126, 169

CommandBuilder object, not instantiating at  
runtime, 121

of DataSource controls, 351–353

DataViews and, 194

of locally persisted DataSets, 209

publications regarding, 96–97

reusing cached query plans for, 122–123

size of DataSets and, 132, 227–229

XML data model and

creating and filling tables, 543–545

populating xml columns, 545–546

tests for, 541–543

updating tables, 546–548

XML indexes and, 546–557

**persistent form objects**

- adding ADO.NET 1.x components to toolbox, 72
- batching multiple updates, 77–79
- comparing ADO.NET 1.x to 2.0, 70–71
- list of, 69–70
- multi-level subforms, 74–75
- parameterized MasterDetailsForm, 75–77
- upgrading to ADO.NET 2.0, 72–73

**pessimistic concurrency control, 189****PIVOT operator, T-SQL, 418, 440–444****Platform Architectural Guidance (PAG) team, 84****polling model, asynchronous SqlCommand execution, 54–55****pooling database connections, 118–120****P&P (patterns and practices), by Microsoft, 84–86. See also best practices****primary keys**

- composite primary keys, 334
- constraint violations, catching, 186–187, 198–201

**PRIMARY XML index, 522, 557****product sales, projecting with linear regression analysis, 484–491****programming, best practices for. See best practices****projects**

- data source-agnostic projects, 40–43
- scalability of, 2
- upgrading to ADO.NET 2.0, 72–73

**properties, usage guidelines for, 103. See also specific properties****PROPERTY XML index, 523, 557****ProviderSpecificDataType property, SqlDataReader object, 45****ProviderType property**

- DataReader object, 45
- DataTableReader object, 45
- SqlDataReader object, 45

**proxy class for Web services, 369–372****Proxy method, SoapHttpClientProtocol class, 371****publications**

- Application Architecture for .NET: Designing Applications and Services*, 84, 85–86
- “Bad Code? FxCop to the Rescue” (Robbins), 110
- Data Patterns*, 84, 87–88
- “Design Guidelines for Class Library Developers”, 102–103
- Designing Data Tier Components and Passing Data Through Tiers*, 85, 97–102
- Distributed Systems Patterns*, 88
- Enterprise Solution Patterns Using Microsoft .NET*, 87
- “Improving .NET Application Performance and Scalability”, 96–97
- Integration Patterns*, 88–89
- “Multiple Active Result Set (MARS) in SQL Server 2005”, 416
- .NET Data Access Architecture Guide*, 85, 96
- “Service-Oriented Integration”, 361
- SQL Server 2005 Books Online, 515
- “Web Services and DataSets” (Skonnard), 107
- Windows Server System Reference Architecture (WSSRA)*, 84
- WS-Addressing specification, 371
- WS-Discovery specification, 371

**Q****queries. See also inserts; updates**

- batch queries, 122
- FOR XML queries, 432
- inspecting with SQL Server Profiler, 120–121
- with no returned data, 12–13
- parameterized, 75–77, 132–133
- recursive queries, 421–422
- returning multiple resultsets, 7–8
- returning resultset as XML stream, 8–10
- returning scalar value, 12
- returning single row, 11–12
- returning tabular data in grid, 10–11
- reusing cached query plans, 122–123
- RPC queries, inspecting, 120–121
- Top n queries, 228

**Query Analyzer, SQL Server, 409****Query Editor, SQL Server, 409**

### query notifications, SQL Server

- automating reorder processing using, 452–453
- definition of, 429–430
- methods for, 444–446
- SqlDependency notifications, 446–448
- SqlNotificationRequest objects, 448–451

### query plans, cached, 122–123

## R

### RangeValidator control, 310, 316–318

### RANK() function, T-SQL, 420–421

### Read method

- SqlDataReader object, 7
- XmlReader object, 9

### ReadCommitted isolation level, 14

### ReadUncommitted isolation level, 14

### Recordset object

- concurrency management by, 189
- disconnected, compared to DataSets, 21–22

### recursive queries, T-SQL, 421–422

### reference architectures, 84, 85–86

### referential integrity, maintaining, 170

### regular expressions, resources for, 315

### RegularExpressionValidator control, 310–311, 315–316, 317–318

### re-hydrating an object, 256

### RepeatableRead isolation level, 14

### Repeater control

- design with XmlDataSource control, 347–348
- SqlDataSource for, 275–279

### Replication Management Objects (RMO), 411

### Report Builder, 79

### Report Manager, SQL Server, 409

### Reporting Services, SQL Server, 79, 409, 412–413

### reports, creating, 79–82, 412–413

### ReportViewer control, 79–82

### RequiredFieldValidator control, 310, 312

### reserved folders in ASP.NET 2.0, 271–272

### ResolveConcurrencyErrors function, 203–208

### resource manager (RM), 217

### resultsets

- multiple, with SqlDataReader objects, 7–8
- returning first column of first row of, 12

returning first row of, 11–12

as XML stream, 8–10

### Resync method, Recordset object, 189

### RetrieveStatistics method,

SqlConnection object, 40, 51–52

### RM (resource manager), 217

### RMO (Replication Management Objects), 411

### Robbins, John (“Bad Code? FxCop to the Rescue”), 110

### RollbackTransaction method,

SqlTransaction object, 13, 14

### ROW\_NUMBER() function, T-SQL, 420–421

### RowValidated event, 184

### RowValidating event, 184

### RPC queries, inspecting, 120–121

### runtime objects, 39–40

## S

### SACT (SQL Server Surface Area Configuration Tool), 408–409

### Saenz-Badillos, Angel (opinions on MARS feature), 415–416

### sales, projecting with linear regression analysis, 484–491

### sample databases

- AdventureWorks sample database
  - compared to Northwind sample database, 133
  - xml columns in, 525–528
- Northwind sample database
  - compared to AdventureWorks sample database, 133
  - configuration change for Access MSDE 2000 server, 7
  - configuration change for SQL Server Express, 39, 133
  - size of, 132

### scalability

- project requirements for, 2
- publications regarding, 96–97

### schema DataTable objects, 43–46

### schema, XML

- adapting to generate a DataSet
  - considerations for, 235–236
  - flat schemas, 245–248
  - nested schemas, 236–245

- flat schema, 245–248
- generating serializable classes from, 255–261
- inferring to generate a DataSet, 248–250
- nested schema, generating DataSet from, 236–245
- SCM (supply chain management), 235**
- Secure Sockets Layer (SSL), 363**
- security, for Web services, 363, 372–374**
- SelectCommand property, DataTable object, 222–224, 225–226**
- serializable classes**
  - creating data sources from, 257–259
  - generating from XML schema, 255–261
- Serializable isolation level, 14**
- Serialization namespace, 255**
- server controls in ASP.NET 2.0, 272**
- server validation controls, 310–311**
- server-side validation, 322–323**
- Service Broker, SQL Server, 425–429**
- Service.asmx file, 363**
- service-oriented architecture (SOA)**
  - definition of, 103
  - history of, 104
  - implementing with Web services, 105–110
- “Service-Oriented Integration”, 361**
- Service.vb file, 363, 364**
- SetSqlDataType method, SqlResultSet object, 21**
- shared memory (Sm), SQL Server, 408**
- shredding, 47**
- SiteMapDataSource control, 273**
- Skonnard, Aaron (“Web Services and DataSets”), 107**
- Slammer/Sapphire worm, 84**
- Sm (shared memory), SQL Server, 408**
- “Smart Client” applications, 131**
- SMO (SQL Management Objects), 410, 411**
- SNAC (SQL Native Client), 414–415**
- Snake/Spida exploit, 84**
- snapshot data, DataTable storing, 189**
- Snapshot isolation level, 14**
- SNAPSHOT isolation level, T-SQL, 418–419**
- SOA (service-oriented architecture)**
  - definition of, 103
  - history of, 104
  - implementing with Web services, 105–110
- SOAP Web services, SQL Server, 430–432, 453–455**
- SOAP-based Web services, 105**
- SoapHttpClientProtocol class, 371, 374**
- SoapVersion method,**
  - SoapHttpClientProtocol class, 371
- software patterns**
  - best practices for, 86–89
  - definition of, 84
- special (reserved) folders in ASP.NET 2.0, 271–272**
- sp\_executesql command?, 122–123**
- SQL Distributed Management Objects (SQL-DMO), 411**
- SQL Management Objects (SMO), 410, 411**
- SQL Mobile scripts, 409**
- SQL Native Client (SNAC), 414–415**
- SQL Profiler, SQL Server**
  - definition of, 412
  - tracing transactions with, 219–220
- SQL queries. See queries**
- SQL Server**
  - Admin connection, 416
  - batch inserts, 47–50
  - BIDS (Business Intelligence Development Studio), 410
  - checking for available instances, 46–47
  - CLR integration, 413, 458
  - connection statistics, 51–52
  - data encryption, 424
  - data types, namespace for, 4
  - Database Mail, 430
  - database mirroring, 416
  - Database Tuning Advisor, 412
  - DDL (Data Definition Language) triggers, 424
  - Developer Edition, 405
  - Enterprise Edition, 406
  - Express Edition, 404–405
  - generating typed DataSet object from, 22–28
  - Management Studio, 409–410
  - MARS (Multiple Active Result Sets), 415–416
  - Mobile Edition, 406
  - namespace for, 4
  - native SOAP Web services, 430–432, 453–455
  - Notification Services, 409, 429

### SQL Server (continued)

#### query notifications

automating reorder processing using,  
452–453

definition of, 429–430

methods for, 444–446

SqlDependency notifications, 446–448

SqlNotificationRequest objects,  
448–451

Reporting Services, 79, 409, 412–413

RMO (Replication Management Objects), 411

SACT (SQL Server Surface Area Configuration  
Tool), 408–409

Service Broker, 425–429

SMO (SQL Management Objects), 410, 411

SNAC (SQL Native Client), 414–415

SQL Profiler, 219–220, 412

SSCM (SQL Server Configuration Manager),  
407–408

SSIS (SQL Server Integration Services), 411

Standard Edition, 406

table partitioning, 416

#### T-SQL

CTEs (common table expressions), 421–422

data encryption, 424–425

DDL (Data Definition Language) triggers, 424

DENSE\_RANK() function, 420–421

FOR XML AUTO clause, 422–424

FOR XML PATH clause, 435–440

FOR XML queries, 432

FOR XML RAW clause, 433–435

new features, list of, 416

PIVOT and UNPIVOT operators, 418,  
440–444

RANK() function, 420–421

recursive queries, 421–422

ROW\_NUMBER() function, 420–421

SNAPSHOT isolation level, 418–419

TABLESAMPLE operator, 419–420

TOP n operator, 419–420

TRY...CATCH blocks, 417–418

version 2000, success of, 403

version 2005

editions in, 404–406

list of new features in, 407

Workgroup Edition, 405–406

xml data type, 413–414, 513–514

XQuery, 413–414, 513–514

**SQL Server CE, namespace for, 4**

**SQL Server Configuration Manager (SSCM),  
407–408**

**SQL Server Integration Services (SSIS), 411**

**SQL Server Management Studio (SSMS),  
409–410**

**SQL Server Mobile Edition, namespace for, 4**

**SQL Server Profiler, 120–121**

#### SQL Server projects

assemblies generated by, 457, 458

attribute decorations for, 458–459

definition of, 457, 458

sample project

assembly for, creating, 466–468

CFS (CLR scalar user-defined function), 462

creating SQL/CLR objects, 468–470

CSP (CLR stored procedure), 461–462

CTR (CLR trigger), 464–465

database connection for, 460

debugging, 471–472

description of, 460, 461

dropping SQL/CLR objects, 470

test scripts for, 465–466, 471–472

UDAs (user-defined aggregates), 463–464

UDTs (user-defined types), 463

templates for, 459

**SQL Server Reporting Services, 79**

**SQL Server Surface Area Configuration Tool  
(SACT), 408–409**

**SQL Server 2000 Best Practices Analyzer  
(BPA), 116–118**

**SQL Server 2005 Books Online, 515**

**SqlBulkCopy object, 40, 47–50**

**SqlBulkCopyOptions enumeration, 48**

**SqlCeClient namespace, 4**

#### SqlClient data provider

bidirectional (navigable) cursors, not  
supported, 6

creating basic data objects, 6–13

**SqlClient namespace, 4**

**SqlClientCe namespace, 4**

**SQL/CLR objects**

attribute decorations identifying, 458–459

CLR integration

definition of, 413

enabling, 458

creating, 468–470

definition of, 457

dropping, 470

examples of, 461–465

performance and, 458

when to use, 457

**SQLCMD mode, SQL Server Management Studio, 410****sql:column function, 513, 538–539****SqlCommand object**

asynchronous execution of

callback model, 55–58

definition of, 40

execution models for, 52–53

polling model, 54–55

WaitAll model, 59–62

definition of, 3

**SqlCommandBuilder object, 121****SqlConnection object**

creating, 24, 26

definition of, 3

statistics from, retrieving, 51–52

**SqlContext object, 461****SqlDataAdapter object**

autoenlisting in implicit transaction, 217–218

manually enlisting in explicit transaction,

220–221

**SqlDataReader object**

with multiple resultsets, 7–8

populating hierarchical objects with, 387–391

properties of, 44–46

**SqlDataRecord object, 473–475****SqlDataSource control**

compared to ObjectDataSource control, 325

creating, 275–279

definition of, 273

**SqlDbType enumeration, 20****SqlDependency notifications, SQL Server, 446–448****SQL-DMO (SQL Distributed Management Objects), 411**

SqlException exception, 181

SQLMail, SQL Server, 430

SqlMetaData object, 473

SqlNotificationRequest object, SQL Server, 448–453

SqlParameter collection, 122

SqlPipe object, 461

<SqlProcedure() > attribute decoration, 458

SqlResultset object, removed from product, 6

SqlResultSet object, typed data from, 19–21

**SqlTableAdapter object**

autoenlisting in implicit transaction, 218–219

manually enlisting in explicit transaction,

220–221

**SqlTransaction object**

multi-table updates, transactions for, 13–16

updating DataSets using, 214

<SqlTrigger> attribute decoration, 458

SqlTypes namespace, 4, 20

<SqlUserDefinedAggregate() > attribute decoration, 458

<SqlUserDefinedFunction() > attribute decoration, 458

<SqlUserDefinedType() > attribute decoration, 458

sql:variable function, 513

SQLXML Managed Classes, 17

SSCM (SQL Server Configuration Manager), 407–408

SSIS (SQL Server Integration Services), 411

SSL (Secure Sockets Layer), 363

SSMS (SQL Server Management Studio), 409–410

standalone DataTable objects, 62–66

Standard Edition, SQL Server, 406

Stored Procedure template, 459

**stored procedures**

batch queries as substitute for, 122

CSP (CLR stored procedure)

creating, 468

definition of, 458, 459

example of, 461–462

examples of, generating XML documents, 475–484

### **stored procedures (continued)**

- examples of, projecting product sales, 484–491
- examples of, returning content-dependent SqlDataRecords, 473–475
- when to use, 472
- reconfiguring DataTableAdapter object to use, 71
- subforms, multi-level, 73–75**
- supply chain management (SCM), 235**
- Surface Area Configuration Tool, SQL Server, 408–409**
- surfacing, 28**
- System.Collections.Generics namespace, 259**
- System.ComponentModel namespace, 3, 259**
- System.Data namespace, 2–4**
- System.Data.Common namespace, 3**
- System.Data.Common.DbConnection namespace, 3**
- System.Data.dll assembly, 3**
- System.Data.Odbc namespace, 4, 5**
- System.Data.OleDb namespace, 4, 5**
- System.Data.OracleClient namespace, 4**
- System.Data.SqlCeClient namespace, 4**
- System.Data.SqlClient namespace, 4**
- System.Data.SqlTypes namespace, 4, 20**
- System.MarshalByRefObject namespace, 3**
- System.Object namespace, 3**
- System.Transactions namespace, 217–221**
- System.XML namespace, 3, 4**
- System.Xml.Serialization namespace, 255**

## **T**

- tabbed forms, migrating data entry form to, 149–153**
- table partitioning, SQL Server, 416**
- table schemas, 43–46**
- TableAdapter object**
  - autogenerating, 132
  - concurrency management by, 190
  - definition of, 70

- parameterized queries in, 75–77
- partial classes for, 228–229
- updatable, commands for, 22
- updates and, 171, 172
- Web services and, 376

### **TableLock member, SqlBulkCopyOptions enumeration, 48**

### **tables**

- adding UDT column to, 499–500
- batch inserts, 47–50
- batch updates, 77–79, 126, 169
- displaying table rows with UDT columns, 500
- multiple, viewing, 32–35, 73–75
- relationships between parent and child tables, 170
- returning values from UDT columns, 501–502
- saving changes to
  - maintaining referential integrity, 170
  - procedure for, 171–180
  - strategy for, 169
- viewing in DataGridView control, 28–30
- viewing in details form, 31–32
- TABLESAMPLE operator, T-SQL, 419–420**
- Tcp (TCP/IP), SQL Server, 408**
- TDD (test-driven development), automating, 112–116**
- TemplateFields, GridView control, 295–297**
- templates**
  - for bound controls, 281–283
  - pattern template, 87
  - SQL Server projects, 459
- TerraService Web service client, 362**
- test scripts, SQL Server projects, 465–466, 471–472**
- test-driven development (TDD), automating, 112–116**
- text boxes**
  - replacing with combo boxes, 156–159
  - validating, 183–184
- Timeout method,**
  - SoapHttpClientProtocol class, 371**
- timestamp column, using for optimistic concurrency control, 123–125**
- ToolTip property, 311**
- TOP n operator, T-SQL, 419–420**

**Top n queries, 228**  
**tracing Web applications, 351–353**  
**Transaction object, 217**  
**Transaction property**  
     SqlTransaction object, 13  
     TableAdapter object, 190  
**TransactionConnection property,**  
     **TableAdapter object, 190**  
**transactions**  
     for concurrency management, 190  
     enlistment of resource manager for, 217–221  
     for multi-table updates, 13–16  
     tracing with SQL Profiler, 219–220  
     updating DataSets in, 214–221  
     with Web services, 360  
**Transactions namespace, 217–221**  
**TransactionScope object, 217–218, 221**  
**TreeView control**  
     populating from XmlDataSource control,  
         348–350  
     SqlDataSource for, 275–279  
**Trigger template, 459**  
**TRY . . . CATCH blocks, T-SQL, 417–418**  
**T-SQL, SQL Server**  
     CTEs (common table expressions), 421–422  
     data encryption, 424–425  
     DDL (Data Definition Language) triggers, 424  
     DENSE\_RANK() function, 420–421  
     FOR XML AUTO clause, 422–424  
     new features, list of, 416  
     PIVOT and UNPIVOT operators, 418  
     RANK() function, 420–421  
     recursive queries, 421–422  
     ROW\_NUMBER() function, 420–421  
     SNAPSHOT isolation level, 418–419  
     TABLESAMPLE operator, 419–420  
     TOP n operator, 419–420  
     TRY . . . CATCH blocks, 417–418  
     when to use, 457  
**typed xml columns**  
     creating XmlSchemaCollection for, 519–520  
     definition of, 519  
     reading XmlSchemaCollection for, 520–522

**U**  
**UBL (Universal Business Language), 236**  
**UDAs (user-defined aggregates)**  
     creating, 413, 469–470  
     definition of, 458, 459  
     example of, 463–464  
**UDDTs (user-defined data types), 491–492**  
**UDFs (user-defined functions), 413**  
**UDTs (user-defined types)**  
     adding UDT column to table, 499–500  
     creating  
         native-format UDT code, 492–494  
         procedure for, 413, 469  
         simple value-type UDT code, 496–498  
         type for, specifying, 492  
         UserDefined-format UDT code, 494–495  
     definition of, 458, 459, 491  
     displaying table rows with UDT columns, 500  
     example of, 463  
     returning values from UDT columns, 501–502  
**Universal Business Language (UBL), 236**  
**UNPIVOT operator, T-SQL, 418, 440–444**  
**Unspecified isolation level, 14**  
**untyped xml columns**  
     definition of, 515  
     populating xml columns, 515–517  
     retrieving and reformatting xml column data,  
         517–518  
**UpdateBatch method, Recordset object,**  
     **189**  
**UpdateBatchSize property**  
     OracleDataAdapter object, 77  
     SqlDataAdapter object, 77  
**updates**  
     batch updates, 77–79, 126, 169  
     cascading updates, 170  
     multi-table, transactions for, 13–16  
     referential integrity, maintaining, 170  
**Url method, SoapHttpClientProtocol**  
     **class, 371**  
**UseInternalTransaction member,**  
     **SqlBulkCopyOptions enumeration, 48**  
**user credentials, for Web service**  
     **authentication, 372–374, 377**

### **UserAgent method,**

**SoapHttpClientProtocol class, 371**

### **user-defined aggregates (UDAs)**

creating, 413, 469–470

definition of, 458, 459

example of, 463–464

### **user-defined data types (UDDTs), 491–492**

### **User-Defined Function template, 459**

### **user-defined functions (UDFs), 413**

### **User-Defined Type template, 459**

### **user-defined types (UDTs)**

adding UDT column to table, 499–500

creating

native-format UDT code, 492–494

procedure for, 413, 469

simple value-type UDT code, 496–498

type for, specifying, 492

UserDefined-format UDT code, 494–495

definition of, 458, 459, 491

displaying table rows with UDT columns, 500

example of, 463

returning values from UDT columns, 501–502

### **UserDefined-format UDT**

accessing data from other fields or tables

using, 509

creating, 494–495

example of, 502–508

XML representation of, 509–511

### **users, disconnected, 182, 208–212**

## **V**

### **Validated event, 183, 184**

### **Validating event, 183, 184**

### **validation**

client-side validation, 126–127, 311, 323–324

for DataGridView control, 184–186

for default values, 187–188

definition of, 181, 183, 310

for GridView control

adding controls to template, 313–315

CompareValidator control for, 318

CustomValidator control for, 319–320

RangeValidator control for, 316–318

RegularExpressionValidator control for, 315–316, 317–318

ValidationSummary control for, 321

primary key constraint violations, catching, 186–187

for text boxes, 183–184

validation groups, 311

Web controls for

list of, 310–311

properties of, 311–312

### **ValidationGroup property, 311**

### **ValidationSummary control, 311, 321**

### **VALUE XML index, 523**

### **VB 2005, migrating to, 1–2**

### **Venus code editor, 265**

### **verbs, FormView control, 292**

### **version control, for Web services, 362**

### **VIA (Virtual Interface Architecture), 408**

### **virtual directory, IIS, deploying Web applications to, 353–356**

### **Visual Basic 6, migrating to Visual Basic 2005, 1–2**

### **Visual Basic 2005, migrating to, 1–2**

### **Visual Studio Upgrade Wizard, 72–73**

### **volatility in database, 189**

### **VSTS suite, 112–116**

### **VWD (Visual Web Developer), 265**

## **W**

### **WaitAll model, asynchronous SqlCommand execution, 59–62**

### **WaitHandle array, 59–62**

### **Web applications**

creating, 266–269

deploying to IIS, 353–356

precompiling, 355–356

tracing, to compare performance, 351–353

### **Web controls. See also specific controls**

for data binding and updates, list of, 272, 273

for data validation, list of, 310–311

### **Web forms. See also bound controls**

compilation model, 269–271

creating, process for, 266–269

scalability requirements, 2

- Web controls
  - for data binding and updates, list of, 272, 273
  - for data validation, list of, 310–311
- Web Matrix, 265**
- Web Reference, adding to configuration files, 362**
- Web services**
  - authentication for, 372–374, 376–377
  - clients
    - DataGridView control for, 380–382
    - proxies for, 369–372
  - connection strings for, 376–377
  - creating
    - example of, 375–380
    - procedure for, 363–364
  - custom business objects
    - as alternative to DataSets, 384–391
    - binding to DataGridViews, 391–395
    - client for, 397–399
    - updating, inserting, deleting, 395–397
    - using, issues regarding, 361
  - DataSets
    - returning, 377–380
    - updating, 382–384
    - using, issues regarding, 361
  - debugging, 374–375
  - deploying, 369
  - development strategies for, 360–363
  - help page for, 364–369
  - history of, 359–360
  - security for, 363, 372–374
  - service-oriented architecture (SOA)
    - implemented using, 105–110
  - transactions and, 360
  - version control, 362
  - WebMethods and, 362
  - Windows Authentication Mode and, 377
  - WSDL document for, 364–369
- “Web Services and DataSets” (Skonnard), 107**
- Web Services Description Language (WSDL), 105**
- Web Services Interoperability (WS-I) Organization, 106**
- Web Site folder, 363**
- Web site resources**
  - AntiPatterns Web site, 86
  - application blocks, 89
  - BPA (Best Practices Analyzer), 116
  - Data Patterns*, 87
  - design guides, 96
  - Distributed Systems Patterns*, 88
  - Electronic Commerce Dictionary, 88
  - Enterprise Solution Patterns Using Microsoft .NET*, 87
  - eWeek article on Gartner market share numbers, 403
  - Global Bank Implementation (GBI) project, 86
  - HR-XML (Human Resources XML), 236
  - Microsoft Architects JOURNAL, 85
  - Microsoft Patterns, 87
  - Microsoft reference architectures, 85
  - MSDN Web Services Developer Center, 106
  - PAG team, 84
  - regular expressions, 315
  - service-oriented architecture (SOA), 103
  - TerraService Web service client, 362
  - UBL (Universal Business Language), 236
  - VSTS suite, 112
  - WSSRA implementation guides, 85
- Web.config file**
  - adding to project, 268
  - adding Web Reference to, 362
  - ConfigurationManager class for, 379
  - connection strings in
    - encryption of, 120
    - how to store, 119, 325, 376
  - enabling anonymous Web service users, 377
  - removing debugging symbols from temporary files, 355, 369
  - specifying multiple traces in, 352
- WebMethods**
  - displaying WSDL document, 364–366
  - invoking from proxy class, 371–372
  - multiple, 362
  - returning typed DataSet, 379–380
- WHERE clause constraints, 283–286**
- white exclamation mark icon, 181**
- Windows Authentication Mode, Web services and, 377**

## **Windows credentials, 373**

### **Windows forms. See also DataGridView control**

- autogenerating
  - features for, 36
  - modifying autogenerated code, 132–133
  - procedure for, 28–30, 32–35
- data entry forms
  - autogenerating, 36, 132–133
  - creating, process for, 70–71
  - database-agnostic applications for, 40
  - heads-down, 146–148
- details form, 31–32
- drag-and-drop methods for, 36, 39–40
- editing forms, creating from XML documents, 250–254
- master-detail forms, 71
- master-detail, parameterized, 75–77
- multi-level subforms, 73–75
- objects for, 39–40
- persistent form objects
  - adding ADO.NET 1.x components to toolbox, 72
  - batching multiple updates, 77–79
  - comparing ADO.NET 1.x to 2.0, 70–71
  - list of, 69–70
  - multi-level subforms, 74–75
  - parameterized MasterDetailsForm, 75–77
  - upgrading to ADO.NET 2.0, 72–73
- tabbed form, migrating data entry form to, 149–153

### **Windows Server System Reference Architecture (WSSRA), 84, 85**

### **Workgroup Edition, SQL Server, 405–406**

### **wrapped nested XML schema, 243–245**

### **WS-Addressing specification, 371**

### **WS-Discovery specification, 371**

### **WSDL document, for Web services, 364–369**

### **WSDL (Web Services Description Language), 105**

### **WS-I (Web Services Interoperability) Organization, 106**

### **WSSRA (Windows Server System Reference Architecture), 84, 85**

## **X**

### **XML constructors, XQuery, 536–538**

#### **XML data model**

- choosing, 515
- indexed `xml` columns, 522–525
- performance and
  - creating and filling tables, 543–545
  - populating `xml` columns, 545–546
  - tests for, 541–543
  - updating tables, 546–548
  - XML indexes and, 546–557
- typed `xml` columns, 519–522
- untyped `xml` columns, 515–518

#### **xml data type, SQL Server**

- definition of, 413–414, 513–514
- indexed `xml` columns
  - add and drop XML indexes, 523–524
  - definition of, 522–523
  - promoting `xml` values to table columns, 524–525
- sample database containing, 525–528
- typed `xml` columns
  - creating `XmlSchemaCollection` for, 519–520
  - definition of, 519
  - reading `XmlSchemaCollection` for, 520–522
- untyped `xml` columns
  - definition of, 515
  - populating `xml` columns, 515–517
  - retrieving and reformatting `xml` column data, 517–518

### **XML DDL, 513, 556–557**

#### **XML documents**

- creating editing forms from, 250–254
- generating with `XmlTextWriter` object, 509–511
- generating with `XmlWriter` object, 475–484

#### **XML files**

- creating `XmlDataSource` control from, 344–345
- populating `GridView` control with, 346
- reading, 343–350

### **XML for Analysis (XML/A) scripts, 409**

### **XML indexes. See indexed xml columns**

### **XML namespace, 3, 4**

**XML schema**

adapting to generate a DataSet  
 considerations for, 235–236  
 flat schemas, 245–248  
 nested schemas, 236–245  
 flat schema, 245–248  
 generating serializable classes from, 255–261  
 inferring to generate a DataSet, 248–250  
 nested schema, generating DataSet from,  
 236–245

**XML/A (XML for Analysis) scripts, 409**

**xmlColumn.exist()** method, 529, 532–533

**xmlColumn.modify()** method, 530,  
 534–535

**xmlColumn.nodes()** method, 530, 533–534

**xmlColumn.query()** method, 529, 530–531

**xmlColumn.value()** method, 529, 531–532

**XmlDataDocument** object, 4

**XmlDataSource** control

creating from XML documents, 344–345  
 definition of, 273  
 designing Repeater control with, 347–348  
 populating TreeView control with, 348–350  
 reading XML files with, 343–350

**XmlReader** object

definition of, 5  
 with FOR XML AUTO clause, 8–10  
 returning with OleDb data provider, 18

**XmlSchemaCollectionDatabase** property,  
**SqlDataReader** object, 46

**XmlSchemaCollectionName** property,  
**SqlDataReader** object, 46

**XmlSchemaCollectionOwningSchema**  
 property, **SqlDataReader** object, 46

**XmlTextWriter** object, generating XML  
 documents with, 509–511

**XmlWriter** object, generating XML  
 documents with, 475–484

**XPath**, 513–514

**XQuery, SQL Server**

definition of, 413–414, 513–514

expressions for xml columns

case-sensitivity, 530

examples of, 530–535

executing with code, 540–541

methods for, 529–530

namespace declarations, 530

syntax for, 528–529

FLWOR expressions, 535–539

XML constructors, 536–538













powered by

books24x7<sup>®</sup>

Programmer to Programmer™



# Take your library wherever you go.

Now you can access more than 70 complete Wrox books online, wherever you happen to be! Every diagram, description, screen capture, and code sample is available with your subscription to the **Wrox Reference Library**. For answers when and where you need them, go to [wrox.books24x7.com](http://wrox.books24x7.com) and subscribe today!

## Find books on

- ASP.NET
- C#/C++
- Database
- General
- Java
- Mac
- Microsoft Office
- .NET
- Open Source
- PHP/MySQL
- SQL Server
- Visual Basic
- Web
- XML



[www.wrox.com](http://www.wrox.com)