

Index

Symbols

- \ (backslash), use in C#, 180
- , (comma), defining parameters in CREATE PROCEDURE statement, 76
- { } (curly brackets), class structure, 22
- “ (quotation marks), delimiting assembly name, 27
- ; (semicolon), C# coding, 24
- [] (square brackets), delimiting assembly name, 27

A

- aborts, 261–262**
- Accumulate() method, 44
- actions trigger, firing, 86**
- Add() overloaded method, 53
- Address Windowing Extensions (AWE), 286**
- Adjust() method, 54–55
- administration, CLR**
 - counters, monitoring with Performance Monitor, 325–326
 - dynamic management views
 - CPU usage, 324
 - loaded assemblies, 322–323
 - memory usage, 323–324
 - monitoring with SQL Server Profiler, 321–322
 - .NET for DBA and System Admin
 - compiled assembly, looking inside, 287–288
 - described, 284
 - memory management, 286
 - server location of .NET Framework, 284
 - SQL Server, CLR version for, 284–286
 - problems facing, 283
 - troubleshooting with SQL Server Profiler
 - database, examining, 327–328
 - described, 326–327
 - Visual Studio debugging, 328–329
- administrator, database**
 - barriers of entry, perspective on, 12
 - deployment cycle, 306
 - new responsibilities, 283
 - Process class, use of, 183
 - source code, advantage of storing with assembly, 31
 - SQL CLR
 - deployment, 19
 - development, locking down, 34
 - maintenance, 13–14
 - TRUSTWORTHY flag, using, 174
 - T-SQL routine, developing and fine-tuning, 19
- ADO for legacy applications, 237–238**
- ADO.NET, connecting to SQL Server, 232**
- AFTER trigger, nesting, 84**
- aggregates, user-defined (UDAs)**
 - altering, 96
 - creating, 96
 - described, 4, 93
 - managed code, 93–96
 - removing, 97
 - SqlUserDefinedAggregate attribute (Format), 96
- aggregation contract, required, 44**
- ALTER PROCEDURE statement, default parameter value, 73–74**
- ALTER TRIGGER statement, 91**
- altered objects database, 86**
- altering managed code, 101**
- ambient transaction**
 - accessing, 66–67
 - described, 62
- application domains, 8**
- application tier for business logic, 11**
- application usage architecture, 231–232**
- applications, installed**
 - design, 227
 - managed procedure to access registry, 228–229
 - needs for, 227
 - results table, storing, 227–228
 - security, importance to, 227, 229–230
- array, T-SQL handling, 125–126**
- ASP web development error-handling method, 258–259**

assembly

- blocking from server, 308–309
- CLR Security, loading
 - described, 291
 - development, restricting, 291–292
- dependencies, reviewing, 39–40
- DLL script loading, 313–314
- dropping, 40
- entry points, DDL to wrap, 316–317
- EXTERNAL_ACCESS permission set, 321
- metadata
 - core assemblies, accessing like extended stored procedures, 304–305
 - noncore assemblies, viewing in catalog, 302–304
- printer, denying access to (Printing permission), 295
- users, permission to load, 309–310
- Visual Studio, 38

assembly level code restricting

- CAS
 - hardening, 300–302
 - model, 299
- CLR permission levels in SQL Server
 - described, 294
 - external permission level, 295–296
 - External_Access permission level, 296–297
 - safe permission level, 295
 - unsafe permission level, 296
- database trustworthiness, 298–299
- described, 293
- outside SQL Server, 294

automatic variables, 106–107

availability, business

- Corp.SQLCLR.Scheduling assembly, 380–381
- SQL CLR solution, 385
- T-SQL solution, 356–357

availability, detecting in-process connections, 49–50

AWE (Address Windowing Extensions), 286

B

backing up

- assemblies, 321
- CLR Security, 321

backslash (\), use in C#, 180

barriers of entry

- DBA perspective, 12
- implementation considerations, 12–13
- maintenance considerations, 13–14
- performance considerations, 13
- security considerations, 11

base types, converting SqlDbTypeypes to, 362

batch, 104

BCL (Base Class Library)

- external data sources, accessing
 - file IO, 201–207
 - Web services, consuming, 213–223
 - XML documents, creating and consuming, 207–213
- monitoring tools, building effective
 - installed software, auditing, 227–230
 - server processes, auditing, 223–227

- namespaces, 24
- server processes, auditing, 223–227
- System.Data assembly, 15
- T-SQL

- capabilities, 137–138

- SQL CLR versus, 10

- unsafe assemblies, testing, 183

BEGIN . . END construct, 107–108

blank (NULL) values

- data type classes, 46–48

- database testing

- for presence of file, 177–178

- for registry data, 181–182

- syntax, 104

- T-SQL, 122–123

- UDFs, 199

business availability

- Corp.SQLCLR.Scheduling assembly, 380–381

- SQL CLR solution, 385

- T-SQL solution, 356–357

business class, Corp.SQLCLR.Scheduling assembly

- defining the rest of, 370–371

business hours

- Corp.SQLCLR.Scheduling assembly

- altering with exceptions, 376–379

- creating and calculating, 371–372

- structure

- altering with exceptions, 354–356

- creating and calculating, 352–353

business logic

- data tier or application tier, 11

- global database use, 11

button point, adding, 246

C

C#

- class, creating, 35

- code structure class file, 43–46

- compiling .NET DLL, 24–25

- described, 12

- files, supported, 320

- flow of control, 107–109

- instance members, 105

- semicolon (;) coding, 24

- stored procedure, creating in same class file, 43–44

calculations

- business availability

- Corp.SQLCLR.Scheduling assembly, 380–381

- SQL CLR solution, 385

- T-SQL solution, 356–357

- date, 233

- T-SQL versus SQL CLR

- data types, 127–128

- loan calculation, 128–130

- two decimal types, adding together, 126–127

calling stack, 268–269

CAS (code access security)

- described, 8–9

- hardening, 300–302

- model, 299
- permission sets, 9, 13
- case study, date calculation**
 - business case, 332–334
 - DateTime calculation
 - described, 334
 - implicit casting issues, 334–335
 - .NET DateTime class, 336–338
 - .NET TimeSpan class, 338–339
 - storing time in date-only fields, 335–336
 - described, 331
 - prerequisites, solution, 342–343
 - solution comparisons
 - deployment and administration, 391
 - described, 386
 - development perspective, 392
 - .NET Console test harness, adding, 388–391
 - performance, 391–392
 - testing and debugging, 386–388
 - tasks outlined, 340–342
- casting**
 - issues, DateTime calculation, 334–335
 - performing, 243
 - T-SQL syntax, 104
- catalogs, SQL Server system functions accessing, 79**
- changes, file system**
 - creating and deleting file, 202–204
 - described, 201
 - design, 202
 - limitations, 207
 - problem, 202
 - updated entry, detecting and updating corresponding file, 204–206
 - verifying, 206
- CIL output, compiler, 5**
- class**
 - CLR routine
 - creating, 34–35
 - deploying, 36–37
 - file, creating stored procedure in same, 43–44
 - stored procedures, 70
 - Visual Studio and structure functions or methods, 38–39
- CLI standard, 4**
- CLR (Common Language Runtime). See also SQL CLR**
 - CLR Objects Throw Errors, 272–274
 - defined, 2
 - processing steps, forward-only, read-only row navigation
 - example, 150
 - routine
 - auto-deployment process, 37
 - creating class, 34–35
 - deploying class, 36–37
 - steps to create, deploy, and use, 18–20
 - routines, listing all, 304
- CLR (Common Language Runtime) administration**
 - counters, monitoring with Performance Monitor, 325–326
 - dynamic management views
 - CPU usage, 324
 - loaded assemblies, 322–323
 - memory usage, 323–324
 - monitoring with SQL Server Profiler, 321–322
 - .NET for DBA and System Admin
 - compiled assembly, looking inside, 287–288
 - described, 284
 - memory management, 286
 - server location of .NET Framework, 284
 - SQL Server, CLR version for, 284–286
 - problems facing, 283
 - troubleshooting with SQL Server Profiler
 - database, examining, 327–328
 - described, 326–327
 - Visual Studio debugging, 328–329
- CLR (Common Language Runtime) Security. See also assembly level code restricting**
 - assemblies, loading
 - db_DDLAdmin database role membership, 293
 - described, 291
 - development, restricting, 291–292
 - role-based user access, 293
 - assembly metadata
 - core assemblies, accessing like extended stored procedures, 304–305
 - noncore assemblies, viewing in catalog, 302–304
 - backups, 321
 - environments
 - development, 305
 - production, 306
 - error-handling stored procedures
 - categories, 257–258
 - clean up after exceptions, 276–277
 - CLR Objects Throw Errors, 272–274
 - at design time, 274–276
 - .NET structured error-handling, 266–270
 - T-SQL @@Error Exception-Handling, 258–262
 - T-SQL object errors, 277–280
 - T-SQL objects handling SQL CLR exceptions, 280–281
 - T-SQL TRY...CATCH error-handling, 262–266
 - .NET framework usage
 - CustomMarshallers.dll assembly, 291
 - described, 288–289
 - Microsoft.VisualBasic.dll assembly, 290
 - Microsoft.VisualBasic.dll assembly, 290
 - mscorlib.dll assembly, 289
 - System.Data.dll assembly, 289–290
 - System.dll assembly, 290
 - System.Security.dll assembly, 291
 - System.Web.Services.dll assembly, 291
 - System.Xml.dll and System.Data.SqlXml.dll assemblies, 290
 - .NET language, limiting, 305
 - problems facing, 283
 - promoting supporting files, 319–321
 - source, promoting through environments
 - addition of supporting files, 314–316
 - assembly entry points, DDL to wrap, 316–317
 - database project, 317–318

CLR (Common Language Runtime) Security (continued)

- deployment folder, set up common, 307–308
- described, 306, 307
- dropping an assembly, scripting, 310–313
- loading an assembly, scripting, 313–314
- permissions, preparing and removing in server environment, 308–310
- scripting overview, 306
- surface area configuration, 288
- Visual Studio deployment, 318–319

CLS agreement, 5

Codd, Dr. E.F. “Ted,” 2

code access security (CAS)

- described, 8–9
- hardening, 300–302
- model, 299
- permission sets, 9, 13

code execution and model, T-SQL versus SQL CLR, 10

code, restricting at assembly level

CAS

- hardening, 300–302
- model, 299

CLR permission levels in SQL Server

- described, 294
- external permission level, 295–296
- External_Access permission level, 296–297
- safe permission level, 295
- unsafe permission level, 296

database trustworthiness, 298–299

- described, 293

outside SQL Server, 294

code reuse, T-SQL forward-only, read-only row navigation example, 153–154

coding, T-SQL

- forward-only, read-only row navigation example, 150–153
- syntax, 104

column

- attributes of given, 54
- computed, 78
- updated, information about, 86

COM (Component Object Model) object, 10. See also OLE object

COM data type, handling, 197–199

comma (,), defining parameters in CREATE PROCEDURE statement, 76

comma-separated value (CSV) file

- console apps, 238–240
- writing data to, 188–190

commingling SQL CLR and T-SQL, 11, 138–140

commit, before SQL CLR transaction, 67

CommitableTransaction class, 65–66

Common Language Runtime. See CLR; CLR administration; CLR Security

compiling

- CIL output adhering to CLI standard, 5
- compiled assembly, looking inside with .NET for DBA and System Admin, 287–288
- .NET DLL, 24–25

- complex types, support for, 10**
- Component Object Model (COM) object, 10. See also OLE object**
- computations. See calculations**
- connecting**
 - objects, cleaning up after exceptions, 276–277
 - to SQL Server, 232
 - SQLExpress database engine problem, 34
- console applications, stored procedures, 238–240**
- constraints, T-SQL performance, 167–168**
- constructor**
 - coding business availability, 381–383
 - described, 62
- conventions, Visual Studio assemblies, 38**
- conversions, SqlDbType types to base types, 362**
- Convert function, points to Varchar data type, 243**
- Corp.Core.Utilities assembly, 360–362**
- Corp.SQLCLR.Scheduling assembly**
 - business availability, calculating, 380–381
 - business class, defining the rest of, 370–371
 - business default hours structure, creating and calculating, 371–372
 - business hours, altering with exceptions, 376–379
 - constructor and driver method, coding business availability, 381–383
 - described, 365
 - exceptions structure, retrieving and calculating, 372–374
 - parameterized time period, considering, 379–380
 - setting up project, 365–367
 - time segments, consolidating for continuity, 374–376
 - TimeSegment class, defining, 367–370
- Corp.SQLCLR.Utilities assembly, 363–364**
- Corp.Utilities, deploying, 364**
- counters**
 - CLR administration, monitoring with Performance Monitor, 325–326
 - log, T-SQL performance, 161
- CPU**
 - SQL Server 2005 SQL CLR support, 14
 - usage, 324
 - Visual Studio 2005 SQL CLR support, 15
- CREATE FUNCTION statement, 80–81**
- CREATE TRIGGER statement, 91**
- CRUD (created, altered, or dropped) database objects, 2–3, 86**
- Cs files, 320**
- Csproj files, 320**
- CSV (comma-separated value) file**
 - console apps, 238–240
 - writing data to, 188–190
- CTS framework, value types and reference types, 4–5**
- curly brackets ({}), class structure, 22**
- cursor**
 - T-SQL processing, 131–136
 - variable-based, designing around, 346–347
- CustomMarshallers.dll assembly, 291**

D**data sources, accessing external**

- file IO, 201–207
- XML documents
 - content, INSERT or UPDATE, 208–210
 - existence, discerning, 208
 - testing solution, 210–212
 - validating, 212–213

data tier for business logic, 11**data type, 44****data type classes**

- conversion, 104
- LOB, 49
- NULL values, 46–48
- SqlTypes versus SqlDbType, 48–49

data type safe, 106**database**

- created, altered, or dropped objects, 86
- retrieving data from, 145–148
- SQL Server Profiler, 327–328
- submitting data to, 141–144
- testing
 - for presence of file, 177–178
 - for registry data, 181–182
- trustworthiness, 298–299
- TRUSTWORTHY flag, 174
- unchecked errors, havoc wreaked by, 260–261

date calculation case study

- business case, 332–334
- DateTime calculation
 - described, 334
 - implicit casting issues, 334–335
 - .NET DateTime class, 336–338
 - .NET TimeSpan class, 338–339
 - storing time in date-only fields, 335–336
- described, 331
- prerequisites, solution, 342–343
- solution comparisons
 - deployment and administration, 391
 - described, 386
 - development perspective, 392
 - .NET Console test harness, adding, 388–391
 - performance, 391–392
 - testing and debugging, 386–388
- tasks outlined, 340–342

date calculations, 233**date data type, 335****DateTime calculation**

- described, 334
- implicit casting issues, 334–335
- .NET DateTime class, 336–338
- .NET TimeSpan class, 338–339
- storing time in date-only fields, 335–336

DBA

- barriers of entry, perspective on, 12
- deployment cycle, 306
- new responsibilities, 283
- Process class, use of, 183

source code, advantage of storing with assembly, 31

SQL CLR

- deployment, 19
- development, locking down, 34
- maintenance, 13–14
- TRUSTWORTHY flag, using, 174
- T-SQL routine, developing and fine-tuning, 19

db_DDLAdmin **database role membership, 293**

DDL, wrapping assembly entry points, 316–317**debugging**

- solution comparisons, 386–388
- Visual Studio with SQL Server Profiler, 328–329

decoration, 35**DELETE statement, 106**

Deleted **tables, managed triggers, 87–88**

dependencies, assembly, 39–40**deployment**

- case study, date calculation, 391
- folder, set up common, 307–308

design bugs, 257**destination table, auditing server processes, 224****deterministic SVFs, 77–78****development**

- CLR Security environments, 305
- locking down, 34
- perspective, date calculation case study, 392
- restricting, 291–292

directory, limit writing to particular, 300–301

Directory Services **permission, 295**

disassembler, IL (Intermediate Language), 287–288**display, SQL Server 2005 SQL CLR support, 14**

Dispose() **method, 59–60**

divide-by-zero error

- design-time checking, 274
- .NET structured error-handling, 269
- SQL CLR error-handling, 270–271

DLL (dynamic link library). See also XPs

- assemblies
 - deploying manually, 306
 - script loading, 313–314
- extracting, 31–32
- naming using project name, 36

DML triggers

- changing (ALTER TRIGGER statement), 91
- creating (CREATE TRIGGER statement), 91
- described, 3, 4, 89–90
- file system changes
 - creating and deleting file, 202–204
 - described, 202
 - design, 202
 - limitations, 207
 - updated entry, detecting and updating corresponding file, 204–206
 - verifying changes, 206
- managed code, 90
- removing (DROP TRIGGER statement), 92
- SqlTrigger attribute, 90–91

DNS **permission, 295**

Doc **files, 320**

documents, XML

- content, INSERT or UPDATE, 208–210
- design, 207
- DOM, 290
- existence, discerning, 208
- problem, 207
- submission, 207
- testing solution, 210–212
- validating, 212–213
- validation, importance of, 207
- driver method, coding business availability, 381–383**
- DROP TRIGGER **statement, 92**
- dropped database objects, 86**
- dropping managed code, 101**
- dynamic link library. See DLL**
- dynamic management views**
 - CPU usage, 324
 - loaded assemblies, 322–323
 - memory usage, 323–324

E

employee data file changes, accommodating. See file system changes

empty (NULL) values

- data type classes, 46–48
- database testing
 - for presence of file, 177–178
 - for registry data, 181–182
- syntax, 104
- T-SQL, 122–123
- UDFs, 199

end date, 333

Enlist **keyword, 61–62**

entry, barriers of

- DBA perspective, 12
- implementation considerations, 12–13
- maintenance considerations, 13–14
- performance considerations, 13
- security considerations, 11

enumeration, 48

enumerator number, 48

Environment Variables **permission, 295**

error ID, aborts listed by, 261–262

error-handling

- ASP web development, 258–259
- clean up after exceptions, 276–277
- CLR
 - Objects Throw Errors, 272–274
 - stored procedures, 257–258
- at design time, 274–276
- .NET structured error-handling, 266–270
- T-SQL
 - described, 118–119
 - @@Error Exception-Handling, 258–262
 - object errors, 277–280
 - objects handling SQL CLR exceptions, 280–281
 - TRY...CATCH error-handling, 262–266
- Event Log **permission, 295**

Event **parameter, DDL trigger, 92**

events, multiple triggers, 85

exception classes, 267

exceptions

- business default hours structure, altering with, 354–356
- clean up after, 276–277
- datetime pairs, parsing, 350–351
- structure, Corp.SQLCLR.Scheduling, 372–374

ExecuteAndSend() **methods, 56–57**

Extended Stored Procedures. See XPs

Extensible Markup Language documents. See XML documents

extension, file

- retrieving files with specified, 187
- supporting files, promoting, 320

external data sources, accessing

- file IO, 201–207
- XML documents
 - content, INSERT or UPDATE, 208–210
 - existence, discerning, 208
 - testing solution, 210–212
 - validating, 212–213

EXTERNAL **permission, 296**

external permission level, 295–296

external resources, accessing, 51–53

External_Access **permission level, 296–297**

EXTERNAL_ACCESS **permission set**

- assemblies
 - backing up, 321
 - blocking from server, 308–309
- described, 9

F

file

- database testing, NULL variable, 177–178
- specified extension, retrieving, 187

File Dialog **permission, 295**

file system changes

- creating and deleting file, 202–204
- described, 201
- design, 202
- limitations, 207
- problem, 202
- updated entry, detecting and updating corresponding file, 204–206
- verifying, 206

FillRow **method, 81–82**

FillRowMethodName, **83**

filtering text file parsing, 252

finally **block, 276–277**

flat-file entries, 149–150

Flessner, Paul (Microsoft's senior vice president, server applications), 1

flow of control, C#, 107–109

folder, setting up common deployment, 307–308

FOR **trigger, nesting, 84**

forward-only, read-only row navigation example, T-SQL

- CLR processing steps, 150
- code reuse, 153–154
- coding, 150–153
- flat-file entries, 149–150
- .NET-based programming technique, 159–161
- performance metrics, 161
- tuning a SQL CLR, 154–159
- UDF processing, 161
- UPDATE statements, 154

functions

- attribute, 35
- data-centric built-in, 119–122
- Visual Studio, 38–39

functions, user-defined (UDF)

- adding, 34–35
- COM data type, handling, 197–199
- forward-only, read-only row navigation example, 161
- method name, 38
- ParseRow function, adding to, 251
- processing
 - forward-only, read-only row navigation example, 161
 - as supported object, 4
- SQL CLR, 32–33
- stored procedures versus, 26–27

G**GC (garbage collector), 286**

- GetInt32() method, 55–56

- GetValue method, 180

- global database use, business logic, 11

H**handling errors**

- ASP web development, 258–259
- clean up after exceptions, 276–277
- CLR
 - Objects Throw Errors, 272–274
 - stored procedures, 257–258
- at design time, 274–276
- .NET structured error-handling, 266–270
- T-SQL
 - described, 118–119
 - @@Error Exception-Handling, 258–262
 - object errors, 277–280
 - objects handling SQL CLR exceptions, 280–281
 - TRY . . . CATCH error-handling, 262–266

hard disk drive

- SQL Server 2005 SQL CLR support, 14
- Visual Studio 2005 SQL CLR support, 15

hardware

- SQL Server 2005 SQL CLR support, 14
- SQLOS, advantages of, 7
- Visual Studio 2005 SQL CLR support, 14–15

- “hello world,” 53–54

hours, business

- Corp.SQLCLR.Scheduling assembly
 - altering with exceptions, 376–379
 - creating and calculating, 371–372
- structure
 - altering with exceptions, 354–356
 - creating and calculating, 352–353

HPAs (Host Protection Attributes), 12**I**

- IEnumerable object, 82

IL (Intermediate Language)

- described, 287
- disassembler, 287–288

implicit transaction

- programming, 58–60
- promoting, 60–61

- InferFromValue() method, 55

- Init() method, 44

in-process connections, 49–50**input parameters, 72**

- Inserted and Deleted tables, accessing, 87–88

installed software, auditing

- design, 227
- managed procedure to access registry, 228–229
- needs for, 227
- results table, storing, 227–228
- security, importance to, 227, 229–230

instance members, 105

- INSTEAD OF triggers, 84

INT scalar results, returning, 27**Intermediate Language (IL)**

- described, 287
- disassembler, 287–288

- Internet, downloading and executing code from. See CAS

- interop namespace, 197

- intranet, downloading and executing code from. See CAS

- isDate() function, 360–361

- Isolated Storage File permission, 295

J

- JIT compiler, .NET architecture, 5

K

- key, registry, 180

- kill method, 184

- Knight, Brian (*Professional SQL Server 2005 Integration Services*), 31

L

- languages, importance of managed, 12–13

- library. See BCL; DLL

- lifetimes, programming transactions, 67

line, ParseRow function, 250

links, 11, 138–140

LOB (large object, handling), 49

local variables, 104

logic, business

- data tier or application tier, 11
- global database use, 11

loop

- resultset, continuous streaming of, 186–187
- T-SQL and .NET, 109–110

M

maintenance

- barriers of entry, 13–14
- T-SQL performance, 168–169

managed aggregates. See UDAs

managed code

- .NET architecture, 6
- SVFs, 78
- TVFs, 81–82
- UDAs, 93–96
- UDTs
 - creating, altering, and dropping, 101
 - described, 97–99
 - Parse() method, 100
 - SqlUserDefinedType attribute, 101
 - ToString() method, 100

managed functions

- described, 4, 77
- SVFs
 - creating (CREATE FUNCTION statement), 80–81
 - described, 77–78
 - managed code, 78
 - parameters, 80
 - SqlFunction attribute, 78–80
- TVFs
 - altering, 83
 - creating, 83
 - dropping, 83
 - managed code, 81–82
 - parameters, 83
 - SqlFunction attribute, 83

managed languages, .NET architecture, 6

managed procedure

- installed software, auditing, 228–229
- server processes, auditing, 224–226

managed stored procedures

- creating, 75–76
- parameters, 72–74
- public static methods, implementing as, 69–70
- return values, 74–75
- SqlProcedure attribute, 70–72
- updating, 77

managed triggers

- described, 83–85
- DML triggers
 - changing (ALTER TRIGGER statement), 91
 - creating (CREATE TRIGGER statement), 91
 - described, 3, 4, 89–90

- managed code, 90

- removing (DROP TRIGGER statement), 92
 - SqlTrigger attribute, 90–91
- enabling and disabling, 88–89
- Inserted and Deleted tables, accessing, 87–88
- SqlTriggerContext class, 86–87

managed types. See UDTs

memory

- .NET for DBA and System Admin management, 286
- SQL Server 2005 SQL support, 14
- usage, viewing, 323–324
- Visual Studio 2005 SQL CLR support, 14

Merge() **method, 44**

Message Queue **permission, 295**

messages

- aborts listed by, 261
- returning from triggers, 86
- SQL CLR
 - attributes of given row and columns, 54
 - ExecuteAndSend() methods, 56–57
 - in-process connections, 50–51
 - overloaded methods, 53–54
 - SqlDataRecord methods, 55–56
 - SqlMetaDataclass methods, 54–55

metadata, SQL Server assembly, 29–30

method

- aggregation contract, required, 44
- name, UDF, 38
- public versus private, 23, 28
- UDTs, required, 45–46
- Visual Studio, 38–39

Microsoft Distributed Transaction Coordinator (MSDTC) service, 60

Microsoft Intermediate Language (MSIL), 4

Microsoft Windows, 14

Microsoft.VisualBasic.dll **assembly, 290**

Microsoft.VisualBasic.dll **assembly, 290**

monitoring tools, server process

- destination table, creating, 224
- managed procedure for monitoring, creating, 224–226
- testing, 226

MSCorLib.dll assembly, 289

MSDTC (Microsoft Distributed Transaction Coordinator) service, 60

MSIL (Microsoft Intermediate Language), 4

multi-column results, T-SQL definition, 27

My Computer membership, 301

N

Name **parameter, SqlTrigger attribute, 90–91**

namespace

- BCL, 24
- described, 24
- .NET, 56
- potentially harmful, 183
- programming transactions (System.Transaction), 58
- SQL CLR, required, 15

System.IO
 creating and deleting file, 202–204
 described, 201
 design, 202
 limitations, 207
 problem, 202
 updated entry, changing and updating corresponding file, 204–206
 verifying changes, 206

naming
 DLL using project name, 36
 timed segment, 332

nesting triggers, 84

.NET
 applications, T-SQL and, 103
 BCL capabilities, 137–138
 CLR Security
 CustomMarshallers.dll assembly, 291
 described, 288–289
 Microsoft.VisualBasic.dll assembly, 290
 Microsoft.VisualBasic.dll assembly, 290
 MSComLib.dll assembly, 289
 System.Data.dll assembly, 289–290
 System.dll assembly, 290
 System.Security.dll assembly, 291
 System.Web.Services.dll assembly, 291
 System.Xml.dll and System.Data.SqlXml.dll assemblies, 290
 Console test harness, 388–391
 date calculations coding, 233
 DateTime class, 336–338
 for DBA and System Admin
 compiled assembly, looking inside, 287–288
 described, 284
 memory management, 286
 server location of .NET Framework, 284
 language, limiting, 305
 namespace, 56
 overloaded methods, 53–54
 programming technique, forward-only, read-only row navigation example, 159–161
 structured error-handling, 266–270
 Studio solution and database projects, 358
 TimeSpan class, 338–339

.NET architecture. See also BCL
 CIL output of compiler adhering to CLI standard, 5
 CLI components listed, 4
 CLS agreement between programming language and class library creators, 5
 CTS framework for value types and reference types, 4–5
 JIT compiler, 5
 managed code and managed languages, 6
 server location, 284
 SQL CLR and, 4
 VES execution environment for CIL code, 5

network shares, downloading and executing code from. See also CAS
 described, 8–9
 permission sets, 9, 13

noncore assemblies, viewing in catalog, 302–304

nondatabase projects overview, 358–360
nondeterministic SVFs, 77–78

NULL values
 data type classes, 46–48
 database testing
 for presence of file, 177–178
 for registry data, 181–182
 syntax, 104
 T-SQL, 122–123
 UDFs, 199

O

object
 errors, handling, 277–280
 SQL choosing, 26–27
 SQL CLR
 exceptions, handling, 280–281
 solution, creating, 383–385
 supported, 3–4
 type decision, T-SQL solution case study, 344

Object Explorer, 29

object-oriented programming (OOP)
 HTML table, generating, 118
 inheritance and polymorphism, 116
 tasks, abstracting into procedures, 117–118
 UDF, 117

ODBC, SQL Server connection, 234–236

office shift, variable time, 341

Oks, Slava (Microsoft SQLOS developer), 7

OLE DB, SQL Server connections, 236

OLE (Object Linking and Embedding) object
 Automation Procedures, SQL CLR versus, 10
 extended stored procedures, 2
 OLE assembly, marking with UNSAFE permission set, 197

one-column calculation, 27

OOP (object-oriented programming)
 HTML table, generating, 118
 inheritance and polymorphism, 116
 tasks, abstracting into procedures, 117–118
 UDF, 117

operating system
 SQL Server 2005 SQL CLR support, 14
 SQLOS, advantages of, 7

optimizers, T-SQL, 111–112

organization, T-SQL
 error handling, 118–119
 object-oriented programming, 116–118
 structured programming, 112–116

output parameters
 managed stored procedures, 72
 registry key, reading, 180

outside data sources, accessing
 file IO, 201–207
 XML documents
 content, INSERT or UPDATE, 208–210
 existence, discerning, 208
 testing solution, 210–212
 validating, 212–213

overflow error

- abort, 261–262
- ignoring, dangers of, 260
- T-SQL calculation procedure, 259–260

overloaded methods, .NET, 53–54

P

parameterized time period

- Corp.SQLCLR.Scheduling assembly, 379–380
- T-SQL solution, 356

parameters

- managed stored procedures, 72–74
- support for, 10
- T-SQL solution case study, 344–345

Parse() method, 100

ParseRow function

- line, breaking up, 250
- UDF, adding to, 251

parsing

- exception datetime pairs, table-valued functions, 350–351
- text file and returning individual items, 82
- timed segments, table-valued functions, 349–350

partial keyword, 23

passed by reference, 180

Pdb files, 320

performance

- barriers of entry, 13
- case study, date calculation, 391–392
- metrics, forward-only, read-only row navigation example, 161
- triggers, 85
- T-SQL
 - comparison methods, 162–167
 - constraints, 167–168
 - counter log, 161
 - maintainability, 168–169
 - portability, 169
 - system views information, 161

Performance Counter permission, 295

Performance Monitor, 325–326

performance review, changing compensation to reflect favorable. See Web services, consuming

permissions

- assemblies, loading, 309–310
- CAS, 13, 300
- preparing and removing in server environment, 308–310

PID (process ID), terminating process by, 185–186

Plnvoke (Platform Invoke service), 193–195

Pipe.Send() method, 54

Point casting, 245–246

point UDT, 241–242

populating DataGridView, 244–245

portability, T-SQL, 169

printing

- debug mode values, 329
- printer, denying assembly access to (Printing permission), 295

procedure code, Web services, 217–218

Process class, use of, 183

process ID (PID), terminating process by, 185–186

processor

- SQL Server 2005 SQL CLR support, 14
- usage, 324
- Visual Studio 2005 SQL CLR support, 15

production, CLR Security, 306

Professional SQL Server 2005 Integration Services (Knight), 31

Profiler, SQL Server

- CLR administration, monitoring with, 321–322
- database, examining, 327–328
- described, 326–327
- Visual Studio debugging, 328–329

program debug database file, supporting, 320

programming transactions

- current transaction, accessing, 66–67
- implicit transactions
 - atomicity, 58–59
 - Dispose() method, calling, 59–60
 - promoting, 60–61
 - reference, adding manually, 58
- lifetimes, 67
- TransactionScope
 - explicit transactions, 65–66
 - options, 62
 - timeout, setting, 62–63

promoting supporting files, CLR Security, 319–321

property

- available for ambient transaction, returning, 66–67
- described, 49

public method, 23, 28

public static methods, 69–70

publishing Web services, 215–216

Q

quotation marks (“), delimiting assembly name, 27

R

RBS/RBI security, 9

read-only access, production machines, 306

recoverable exceptions, 257

reference type

- CTS framework, 4–5
- value type versus, 44

references

- incorrect properties, TVFs, 78
- manually adding, 58
- Web services, 216–217

registry. See also installed software, auditing

- assembly access to, controlling, 295
- data, database testing, 181–182
- installed software, auditing, 228–229
- key, reading, 179

relational databases, havoc wreaked by unchecked errors, 260–261

release documentation files, 320

removing objects

- dropping assembly, 40
- reviewing assembly dependencies, 39–40

Report Server Project Wizard, 255–256

results table, storing, 227–228

resultset

- continuous streaming of, 186–187
- returning from triggers, 85, 86
- SQL CLR
 - attributes of given row and columns, 54
 - ExecuteAndSend() methods, 56–57
 - in-process connections, 50–51
 - overloaded methods, 53–54
 - SqlDataRecord methods, 55–56
 - SqlMetaDataclass methods, 54–55

resumes, handling. See XML (Extensible Markup Language) documents

retrieving data from database, 145–148

return values, managed stored procedures, 74–75

role-based user access, 293

rollback before SQL CLR transaction, 67

root, registry key, 179

routine

- CLR, listing all, 304
- user-defined functions versus stored procedures, 26–27

row, table

- attributes of given, 54
- forward-only, read-only row navigation example, T-SQL
 - CLR processing steps, 150
 - code reuse, 153–154
 - coding, 150–153
 - flat-file entries, 149–150
 - .NET-based programming technique, 159–161
 - performance metrics, 161
 - tuning a SQL CLR, 154–159
 - UDF processing, 161
 - UPDATE statements, 154
- high-value, importing only, 253–254
- parsing into file stream with comma delimiter, 189–190
- populating, 83, 187

S

SAFE permission set, 9

scalar results, returning, 27

Scalar-Valued Functions (SVFs)

- creating (CREATE FUNCTION statement), 80–81
- described, 77–78
- managed code, 78
- parameters, 80
- SqlFunction attribute, 78–80

scheduling assembly

- business availability, calculating, 380–381
- business class, defining the rest of, 370–371
- business default hours structure, creating and calculating, 371–372

business hours, altering with exceptions, 376–379

constructor and driver method, coding business availability, 381–383

described, 365

exceptions structure, retrieving and calculating, 372–374

parameterized time period, considering, 379–380

setting up project, 365–367

time segments, consolidating for continuity, 374–376

TimeSegment class, defining, 367–370

scheduling server auditing, 227

scoping variables within error-catching blocks, 270

script loading, 313–314

security

barriers of entry, 11

CAS

described, 8–9

hardening, 300–302

model, 299

permission sets, 9, 13

installed software, auditing, 227, 229–230

SQL CLR model

CAS permission sets, 9

RBS/RBI, 9

SQL injection, 189, 245–246

SQL Server external resources, accessing, 51–53

security, CLR. See also assembly level code restricting

assemblies, loading

db_DDLAdmin database role membership, 293

described, 291

development, restricting, 291–292

role-based user access, 293

assembly metadata

core assemblies, accessing like extended stored

procedures, 304–305

noncore assemblies, viewing in catalog, 302–304

backups, 321

environments

development, 305

production, 306

error-handling stored procedures

categories, 257–258

clean up after exceptions, 276–277

CLR Objects Throw Errors, 272–274

at design time, 274–276

.NET structured error-handling, 266–270

T-SQL @@Error Exception-Handling, 258–262

T-SQL object errors, 277–280

T-SQL objects handling SQL CLR exceptions, 280–281

T-SQL TRY...CATCH error-handling, 262–266

.NET framework usage

CustomMarshallers.dll assembly, 291

described, 288–289

Microsoft.VisualBasic.dll assembly, 290

Microsoft.VisualBasic.dll assembly, 290

MSCorLib.dll assembly, 289

System.Data.dll assembly, 289–290

System.dll assembly, 290

System.Security.dll assembly, 291

security, CLR (continued)

.NET framework usage (continued)

- System.Web.Services.dll assembly, 291
- System.Xml.dll and System.Data.SqlXml.dll assemblies, 290

.NET language, limiting, 305

problems facing, 283

promoting supporting files, 319–321

source, promoting through environments

- addition of supporting files, 314–316

- assembly entry points, DDL to wrap, 316–317

- database project, 317–318

- deployment folder, set up common, 307–308

- described, 306, 307

- dropping an assembly, scripting, 310–313

- loading an assembly, scripting, 313–314

- permissions, preparing and removing in server environment, 308–310

- scripting overview, 306

surface area configuration, 288

Visual Studio deployment, 318–319

Security permission, **295, 296**

SELECT statement

- CSV file, writing data to, 188–190

- logic dependent on results of, 106

semicolon (;), C# coding, 24

serialization assembly, consuming Web services, 221

server location, .NET Framework, 284

server processes, auditing

- design, 223

- destination table, creating, 224

- managed procedure for monitoring, creating, 224–226

- problem, 223

- scheduling, 227

- testing, 226

services, Web

- calling and passing to T-SQL procedure, 218–220

- described, 213

- design, 214

- procedure code to implement, 217–218

- publishing to location where can be found and consumed, 215–216

- reference, adding and naming, 216–217

- review, 214

- scheduling, 223

- serialization assembly, 221

- testing, 220, 221–222

- verifying, 222

SET statement, declaring T-SQL variable, 105

set-based operations, T-SQL, 148–149

severity codes, trapping errors by, 261–262

size limit, UDA, 93

Socket Access permission, **296**

software

- application domains, 8

- application tier for business logic, 11

- application usage architecture, 231–232

software, auditing installed

- design, 227

- managed procedure to access registry, 228–229

- needs for, 227

- results table, storing, 227–228

- security, importance to, 227, 229–230

source, CLR Security

- addition of supporting files, 314–316

- assembly entry points, DDL to wrap, 316–317

- database project, 317–318

- deployment folder, set up common, 307–308

- dropping an assembly, scripting, 310–313

- loading an assembly, scripting, 313–314

- permissions, preparing and removing in server environment, 308–310

source code

- DBA, advantage of storing with assembly, 31

- extracting, 31–32

sp_addextendedproperty system stored procedure, 91

sp_oo Procedures, 2

SQL Client, SQL Server connection, 236–237

SQL CLR

- aggregates, 246–248

- application domains, 8

- architecture, 7–8

- availability, detecting, 49–50

- barriers of entry

 - DBA perspective, 12

 - implementation considerations, 12–13

 - maintenance considerations, 13–14

 - performance considerations, 13

 - security considerations, 11

- business availability, calculating, 385

- C# class file code structure, 43–46

- Corp.Core.Utilities assembly, 360–362

- Corp.SQLCLR.Scheduling assembly

 - business availability, calculating, 380–381

 - business class, defining the rest of, 370–371

 - business default hours structure, creating and calculating, 371–372

 - business hours, altering with exceptions, 376–379

 - constructor and driver method, coding business availability, 381–383

 - described, 365

 - exceptions structure, retrieving and calculating, 372–374

 - parameterized time period, considering, 379–380

 - setting up project, 365–367

 - time segments, consolidating for continuity, 374–376

 - TimeSegment class, defining, 367–370

- Corp.SQLCLR.Utilities assembly, 363–364

- Corp.Utilities, deploying, 364

- data access

 - external resources, accessing, 51–53

 - in-process connections, 50–51

- data type classes

 - LOB, 49

 - NULL values, 46–48

 - SqlTypes versus SqlDbType, 48–49

- described, 2–3, 357–358

- evolution, 2–3

- hosting, 7

- key decisions
 - data tier or application tier for business logic, 11
 - Extended Stored Procedures versus SQL CLR, 10
 - OLE Automation Procedures versus SQL CLR, 10
 - T-SQL versus SQL CLR, 9–10
- namespaces, required, 15
- .NET architecture and, 4
- .NET Studio solution and database projects, 358
- nondatabase projects overview, 358–360
- object, creating, 383–385
- objects, supported, 3–4
- programming transactions
 - current transaction, accessing, 66–67
 - implicit transaction promoting, 60–61
 - implicit transactions, 58–60
 - lifetimes, 67
 - TransactionScope explicit transactions, 65–66
 - TransactionScope options, 62
 - TransactionScope timeout, setting, 62–63
- removing objects
 - dropping assembly, 40
 - reviewing assembly dependencies, 39–40
- resultsets and messages
 - attributes of given row and columns, 54
 - ExecuteAndSend() methods, 56–57
 - overloaded methods, 53–54
 - SqlDataRecord methods, 55–56
 - SqlMetaDataclass methods, 54–55
- return type, importance of, 74–75
- routine
 - assembly, creating, 25–26
 - coding class, 22–24
 - .NET DLL, compiling, 24–25
 - T-SQL stored procedure, 27–28
 - user-defined functions versus stored procedures, 26–27
- security model
 - CAS permission sets, 9
 - RBS/RBI, 9
- SQL Server 2005 support, 14, 20–21
- TVFs, 248–249
- user-defined function, 32–33
- Visual Studio 2005 SQL support, 14–15
- SQL Server**
 - assembly, managing
 - described, 28–29
 - DLL and source code, extracting, 31–32
 - metadata, 29–30
 - catalogs, functions accessing, 79
 - CLR version for, 284–286
 - connections, stored procedure
 - ADO for legacy applications, 237–238
 - ODBC, 234–236
 - OLE DB, 236
 - SQL Client, 236–237
 - T-SQL versus SQL CLR, 10
- SQL Server Express, 18**
- SQL Server Integration Services (SSIS)**
 - data sample, 248–249
 - using TVF in SSIS package, 253–254
- SQL Server Management Studio (SSMS), Microsoft, 21, 75**
- SQL Server Profiler**
 - CLR administration, monitoring with, 321–322
 - database, examining, 327–328
 - described, 326–327
 - Visual Studio debugging, 328–329
- SQL Server's operating system (SQLOS), 7**
- SQL (Structured Query Language)**
 - connection objects, cleaning up after exceptions, 276–277
 - foundations of, 2–3
 - injection, security concerns about, 189, 245–246
 - object, choosing, 26–27
 - SELECT statement
 - CSV file, writing data to, 188–190
 - logic dependent on results of, 106
- SqlConnection **permission, 295, 296**
- SqlContext **object, 53**
- SqlDataRecord **methods, 55–56**
- SqlDataTypes, **362**
- SqlDbType, **48–49**
- SQLExpress database engine connection problem, 34**
- SqlFunction **attribute**
 - described, 78–80
 - TVFs, 83
- SqlMetaDataclass **methods, 54–55**
- SQLOS (SQL Server's operating system), 7**
- SqlPipe **object, 53–54**
- SqlProcedure **attribute, 70–72**
- SqlTriggerContext **class, 86–87**
- SqlTypes, **48–49, 73**
- SqlUserDefinedAggregate **attribute (Format), 96**
- SqlUserDefinedType **attribute, 101**
- square brackets ([]), delimiting assembly name, 27**
- SSIS (SQL Server Integration Services)**
 - data sample, 248–249
 - using TVF in SSIS package, 253–254
- SSMS (SQL Server Management Studio), Microsoft, 21**
- start date, 333**
- static routines, 23–24**
- stored information, system view, 29**
- stored procedure. See also triggers**
 - class, defined, 70
 - console applications and extended procedures, 238–240
 - creating in same class file, 43–44
 - described, 3, 4
 - logic dependent on results of UPDATE, DELETE, or SELECT statement, 106
 - method name, 39
 - Reporting Services and SQL CLR TVFs, 255–256
 - SQL Server connections
 - ADO for legacy applications, 237–238
 - described, 232–234
 - ODBC, 234–236
 - OLE DB, 236
 - SQL Client, 236–237

stored procedure (continued)

- SSIS
 - building TVF, 249–253
 - data sample, 248–249
 - using TVF in SSIS package, 253–254
- user-defined functions versus, 26–27
- Web forms and SQL CLR aggregates, 246–248
- WinForms and SQL CLR UDTs
 - accepting WinForm input to save, 245–246
 - point UDT, creating, 241–242
 - Windows form to display, 242–245
- storing**
 - results table, 227–228
 - source code with assembly, 31
 - time in date-only fields, 335–336
- streaming, continuous resultset, 186–187**
- string**
 - converting, 362
 - length, table-valued functions, 348–349
 - parsing in T-SQL, 123–125
- structured programming, T-SQL**
 - cursors, 114–115
 - HTML page, displaying, 113–114
 - libraries, 116
 - organizing, 112–113
 - stored procedures, 115–116
- Structured Query Language. See SQL**
- structures, Visual Studio, 38**
- submitting data to database, 141–144**
- surface area configuration, CLR Security, 288**
- SVFs (Scalar-Valued Functions)**
 - creating (CREATE FUNCTION statement), 80–81
 - described, 77–78
 - managed code, 78
 - parameters, 80
 - SqlFunction attribute, 78–80
- syntax checking, T-SQL, 111**
- `sys.assemblies` **view, 29–30**
- system functions, T-SQL syntax, 106–107**
- system processes, turning on and off (Process class), 183**
- system view**
 - CPU usage, 324
 - creating with SQL Server metadata, 304
 - stored information, 29
 - `sys.assemblies`, 29–30
 - T-SQL performance, 161
- `System.Data` **assembly, 15**
- `System.Data.dll` **assembly, 289–290**
- `System.Data.SqlXml.dll` **assembly, 290**
- `System.dll` **assembly, 290**
- `System.IO` **namespace, file system access**
 - creating and deleting file, 202–204
 - described, 201
 - design, 202
 - limitations, 207
 - problem, 202
 - updated entry, changing and updating corresponding file, 204–206
 - verifying changes, 206

- `System.Security.dll` **assembly, 291**
- `System.Transaction`, **58**
- `System.Web.Services.dll` **assembly, 291**
- `System.Xml.dll` **assembly, 290**

T

table

- column
 - attributes of given, 54
 - computed, 78
 - updated, information about, 86
- definition, 83
- row
 - attributes of given, 54
 - high-value, importing only, 253–254
 - parsing into file stream with comma delimiter, 189–190
 - populating, 83, 187
- table-valued functions
 - exception datetime pairs, parsing, 350–351
 - string length, 348–349
 - timed segments, parsing, 349–350

Table-Valued Functions (TVFs)

- altering, 83
- creating, 83
- dropping, 83
- managed code, 81–82
- parameters, 83
- referencing incorrect properties, 78
- SqlFunction attribute, 83

Target parameter, DDL trigger, 92

Template Explorer, 75

`Terminate()` **method, 44**

terminating process by PID, 185–186

testing

- BCL for unsafe assemblies, 183
- .NET Console test harness, 388–391
- server processes, auditing, 226
- solution comparisons, 386–388
- Web services, consuming, 220, 221–222

text file, parsing

- returning individual items, 82
- SSIS processing
 - building TVF, 249–253
 - described, 248–249
 - using TVF, 253–254

time data type, 335

time segments, consolidating

- `Corp.SQLCLR.Scheduling` **for continuity, 374–376**

timed segment

- naming, 332
- parsing table-valued functions, 349–350

timeout parameter, 62–63

`TimeSegment` **class, 367–370**

timestamp data type, 335

`ToString()` **method, 52–53, 100**

- TransactionScope
 - class, 60
 - explicit transactions, 65–66
 - timeout, setting, 62–63
- triggers**
 - changing (ALTER TRIGGER statement), 91
 - creating (CREATE TRIGGER statement), 91
 - described, 3, 4, 89–90
 - file system changes
 - creating and deleting file, 202–204
 - described, 202
 - design, 202
 - limitations, 207
 - updated entry, detecting and updating corresponding file, 204–206
 - verifying changes, 206
 - firing, 86
 - managed code, 90
 - messages, returning from, 86
 - nesting, 84
 - performance, 85
 - removing (DROP TRIGGER statement), 92
 - SqlTrigger attribute, 90–91
 - Visual Studio, generated T-SQL, 91
- troubleshooting with SQL Server Profiler**
 - database, examining, 327–328
 - described, 326–327
 - Visual Studio debugging, 328–329
- TRUSTWORTHY flag**
 - assemblies, backing up, 321
 - described, 174
 - warning when it's off, 298–299
- TRY...CATCH error-handling**
 - batch statement abort, 263–264, 265
 - described, 262–263
 - Error_Number() function, 265–266
 - print statement, 264
- TryParse() method, 360–361**
- T-SQL**
 - capabilities
 - array handling, 125–126
 - commingling SQL CLR and T-SQL, 11, 138–140
 - computations and calculations, 126–130
 - cursor processing, 131–136
 - data-centric built-in functions, 119–122
 - .NET BCL capabilities, 137–138
 - NULL values, handling, 122–123
 - retrieving data from database, 145–148
 - set-based operations, 148–149
 - string parsing, 123–125
 - submitting data to database, 141–144
 - cursor processing, 131–136
 - exception handling
 - @@Error, 258–262
 - TRY...CATCH, 262–266
 - forward-only, read-only row navigation example
 - CLR processing steps, 150
 - code reuse, 153–154
 - coding, 150–153
 - flat-file entries, 149–150
 - .NET-based programming technique, 159–161
 - performance metrics, 161
 - tuning a SQL CLR, 154–159
 - UDF processing, 161
 - UPDATE statements, 154
 - .NET applications and, 103
 - organization
 - error handling, 118–119
 - object-oriented programming, 116–118
 - structured programming, 112–116
 - performance
 - comparison methods, 162–167
 - constraints, 167–168
 - counter log, 161
 - maintainability, 168–169
 - portability, 169
 - system views information, 161
 - performance comparison methods, 162–167
 - retrieving data from database, 145–148
 - set-based operations, 148–149
 - solution case study
 - business availability, calculating, 356–357
 - business default hours structure, altering with exceptions, 354–356
 - business default hours structure, creating and calculating, 352–353
 - described, 343–344
 - object type decision, 344
 - parameterized time period, 356
 - table-valued functions, creating, 345–351
 - validating date and time parameters, 345–346
 - submitting data to database, 141–144
 - syntax
 - automatic variables and system functions, 106–107
 - coding, 104
 - controls, flow of, 107–111
 - custom attributes and optimizers, 111–112
 - syntax checking, 111
 - variable declaration and scope, 104–106
 - Web services, calling and passing to, 218–220
- TVFs (Table-Valued Functions)**
 - altering, 83
 - creating, 83
 - dropping, 83
 - managed code, 81–82
 - parameters, 83
 - referencing incorrect properties, 78
 - SqlFunction attribute, 83
- Txt files, 320**
- type, 44**
- type classes**
 - conversion, 104
 - LOB, 49
 - NULL values, 46–48
 - SqlTypes versus SqlDbType, 48–49
- type safe, 106**

types, user-defined (UDTs)

- described, 4, 97
- managed code
 - creating, altering, and dropping, 101
 - described, 97–99
 - Parse() method, 100
 - SqlUserDefinedType attribute, 101
 - ToString() method, 100
- methods, required, 45–46
- X and Y coordinates, accepting to store in SQL Server, 245–246

U

UDAs (user-defined aggregates)

- altering, 96
- creating, 96
- described, 4, 93
- managed code, 93–96
- removing, 97
- SqlUserDefinedAggregate attribute (Format), 96

UDF (user-defined function)

- adding, 34–35
- COM data type, handling, 197–199
- forward-only, read-only row navigation example, 161
- method name, 38
- ParseRow function, adding to, 251
- processing
 - forward-only, read-only row navigation example, 161
 - as supported object, 4
- SQL CLR, 32–33
- stored procedures versus, 26–27

UDTs (user-defined types)

- described, 4, 97
- managed code
 - creating, altering, and dropping, 101
 - described, 97–99
 - Parse() method, 100
 - SqlUserDefinedType attribute, 101
 - ToString() method, 100
- methods, required, 45–46
- X and Y coordinates, accepting to store in SQL Server, 245–246

unrecoverable exceptions, 257

unsafe assemblies, testing BCL, 183

unsafe permission level, 296

UNSAFE permission set

- assemblies
 - backing up, 321
 - blocking from server, 308–309
- described, 9
- listing, 190
- OLE assembly, marking, 197

unwound calling stack, 268–269

UPDATE statements

- forward-only, read-only row navigation example, 154
- logic dependent on results of, 106

updated columns, information about, 86

updating data or schema, T-SQL definition, 27

URL, Yahoo! stock retrieval service, 191–193

user errors, 257

User Interface permission, 296

user-defined aggregates. See UDAs

user-defined function. See UDF

user-defined types. See UDTs

users, controlling access. See permissions

using statement, 35

V

values

- name, reading registry key, 180
- NULL
 - data type classes, 46–48
 - database testing, 177–178, 181–182
 - syntax, 104
 - T-SQL, 122–123
 - UDFs, 199
- type
 - CTS framework, 4–5
 - reference type versus, 44

variable

- cursors, designing around, 346–347
- declaration and scope, 104–106
- scoping within error-catching blocks, 270

Vb files, 320

VB.NET

- barrier of entry, perspective, 12
- compiling .NET DLL, 24–25
- files, supported, 320

Vbproj files, 320

verifying

- file system changes, 206
- Web services, 222

VES execution environment for CIL code, 5

view, system

- CPU usage, 324
- creating with SQL Server metadata, 304
- stored information, 29
- sys.assemblies, 29–30
- T-SQL performance, 161

Visual Studio

- CLR routine, creating
 - auto-deployment process, 37
 - creating class, 34–35
 - deploying class, 36–37
 - described, 33–34
- conventions
 - assemblies, 38
 - class and structure functions or methods, 38–39
 - classes, 38
 - structures, 38
- debugging with SQL Server Profiler, 328–329
- deployment, CLR Security, 318–319
- trigger, generated T-SQL, 91

Visual Studio 2005 SQL CLR support

- hardware, 14–15
- memory, 14

W

Web Access **permission, 296**

Web forms, 246–248

Web services, consuming

calling and passing to T-SQL procedure, 218–220

described, 213

design, 214

procedure code to implement, 217–218

publishing to location where can be found and consumed, 215–216

reference, adding and naming, 216–217

review, 214

scheduling, 223

serialization assembly, 221

testing, 220, 221–222

verifying, 222

weekends, 354, 377–378

WHERE clause, 252–253

Windows (Microsoft), 14

WinForms and SQL CLR UDTs (user-defined types)

accepting WinForm input to save, 245–246

point UDT, creating, 241–242

Windows form to display, 242–245

WITH PERMISSION SET argument, 296–297

worms, detecting. See server processes, auditing

X

X and Y coordinates, accepting to store in SQL Server, 245–246

X509 Store **permission, 296**

XML (Extensible Markup Language) documents

content, INSERT or UPDATE, 208–210

design, 207

DOM, 290

existence, discerning, 208

problem, 207

submission, 207

testing solution, 210–212

validating, 212–213

validation, importance of, 207

xmisp_regread **procedure, 181–182**

xp_fileexist, **175**

xp_regread **extended procedure, 179**

XPs (Extended Stored Procedures)

console applications, 238–240

described, 2

SQL CLR versus, 10

Y

Y and X coordinates, accepting to store in SQL Server, 245–246

Yahoo! stock retrieval service, 191–193

Z

zero, divide-by-error

design-time checking, 274

.NET structured error-handling, 269

SQL CLR error-handling, 270–271