

# Index

## A

**A NULL handle has been used error, 271**

**abstract attribute, 172**

**Abstract Class type, 103**

**abstract objects, 161**

**abstract relationships, 161**

Access **attribute, 179**

**access to work items, controlling, 518**

**ACT (Application Center Test), 394**

**activities**

Microsoft Solutions Framework (MSF), 578

process guidance customization

adding steps to, 598

assigning roles to, 598

creating new activities, 597–598

referencing work products, 597

workstreams, associating activities to,  
598

process templates, 597–598

**Additional files to be deployed with this  
generic test option, 460**

**administering project details**

deleting a Team project, 514–516

overview, 505

project alerts, 511–513

project security management, 507–508

project settings, configuring

overview, 507

project security management, 507–508

structuring your project, 508–510

Team Project Portal, 511

version control management, 510–511

structuring your project, 508–510

Team Foundation Server, connecting to,  
505–506

Team Portal customization and extensibility,  
513–514

Team Project Portal, 511

version control management, 510–511

**administrator role, 543**

**advocacy, 584–585**

**Agile Management for Software Engineering  
(Anderson), 572**

**Agile movement, 568–569**

AllItemsAreInstancesOfType **method,**  
**374**

AllItemsAreNotNull **method, 374**

AllItemsAreUnique **method, 374**

**Allocation View for session reports, 328**

**ALM (Application Lifecycle Model), 569**

**analyzing ordered tests, 354–355**

**anatomy of a work item, 516–517**

**Anderson, David J.**

*Agile Management for Software Engineer-*  
*ing, 572*

blog, 570

**annotation support**

in C, 254–255

in C++, 254

### **annotation support (continued)**

overview, 253–254

SA\_FormatString attribute, 254

SA\_InvalidCheck attribute, 254

SA\_No attribute, 254

SA\_NoAccess attribute, 254

SA\_Post attribute, 254

SA\_Pre attribute, 254

SA\_Read attribute, 254

SA\_ReadWrite attribute, 254

SA\_Write attribute, 254

SA\_Yes attribute, 254

### **Application Center Test (ACT), 394**

#### **Application Designer**

copying diagrams, 54

defining default deployment from

binding applications to servers, 85–87

overview, 83–85

prohibited bindings, 87

System View, 85

validating the deployment, 87–88

described, 13–14

implementation of applications

DealingApp implementation, 50–52

MarketMaker implementation, 43–45

overview, 41–42

SDM files, 42–43

StockBroker implementation, 45–48

StockQuoteApp implementation, 48–50

limitations and workarounds

console applications, 56

DCOM applications, 55

dynamic modeling, 56–57

.NET Remoting, 55

overview, 54

Windows applications and services, 56

model and code synchronization, 52

overview, 30

printing diagrams, 54

reverse engineering, 53

Settings and Constraints Editor, 53–54

Stockbroker example

database connection properties, specifying,  
36–37

design characteristics, 33–34

drawing the diagram, 34–36

endpoint details, specifying, 37–38

endpoint prototype, defining your own,  
38–39

existing service, representing an, 39–40

overview, 32–33

UML comparisons, 34

toolbox

applications, 31–32

ASPNETWebapplication, 32

ASPNETWebService, 32

BizTalkWebService, 32

comment tool, 31

connection tool, 31

endpoints, 31

ExternalDatabase, 32

ExternalWebService, 32

General Designer, 31

GenericApplication, 32

GenericEndpoint, 31

OfficeApplication, 32

overview, 30

pointer tool, 31

WebContentEndpoint, 31

WebServiceEndpoint, 31

WindowsApplication, 32

### **Application layer endpoint, 196–198**

### **Application Lifecycle Model (ALM), 569**

#### **application time, 322**

#### **Application Verifier**

configuration, 264–266

Conserve Memory setting, 264

handle troubleshooting

The current thread is performing a wait

operating on a thread handle in DIIMain  
error, 271

The handle used for the current operation is  
invalid error, 270

Incorrect object type for handle error, 271

Invalid parameters passed to a multi-object  
wait function error, 270

Invalid TLS index in the current operation  
error, 271

- A NULL handle has been used error, 271
- overview, 270
- heap troubleshooting
  - Attempt to destroy process heap error, 269
  - Corrupted end stamp of block header error, 270
  - Corrupted infix pattern for freed block error, 269
  - Corrupted prefix pattern error, 270
  - Corrupted start stamp of block header error, 270
  - Corrupted suffix pattern error, 269
  - Exception raised while verifying block header error, 269
  - Excessive size for the current operation error, 268
  - Full page heap, 267
  - The heap block object of the current operation is corrupted error, 268–269
  - Heap handle with incorrect signature error, 268
  - Heap operation performed on an invalid heap handle error, 268
  - Light page heap, 267
  - Memory Access Operation in the context of a freed block reuse-after-delete or double-delete error, 267
  - Memory Access Operation in the context of an allocated block: heap overrun or heap underrun error, 267
  - Multithreaded access in a
    - HEAP\_NO\_SERIALIZE heap error, 267–268
  - overview, 266–267
  - Process heap list count is wrong error, 270
  - Unexpected exception raised in heap code path error, 269
- how it works, 274–278
- lock troubleshooting
  - Critical section over-released or corrupted error, 272
  - The lock count is invalid error, 273
  - The Lock has already been initialized error, 272
  - The Lock has been double initialized error, 272
  - The lock has been over released error, 273
  - The lock is being used without having been initialized error, 273
  - The lock is corrupted error, 272–273
  - Module contains an active lock error, 272
  - overview, 271–272
  - The owner of the lock is invalid, owner was expected error, 273
  - The recursion count is invalid error, 273
  - The region of memory about to be freed contains an active lock error, 273
  - Released memory containing an active lock error, 272
  - Released virtual memory containing an active lock error, 272
  - The thread cannot own a lock in this context error, 272
  - Thread not supposed to be owning a lock error, 273
  - overview, 263–264
  - programming, 278–281
  - Protection Location setting, 265
  - setup, 264–266
  - Verification Layer settings, 266
  - versions, 263
- ApplicationHost layer endpoint, 199–200**
- architect role, 587–588**
- architecting your Active Directory (AD) structure, 653–654**
- architects and MSF for CMMI Process Improvement, 571**
- architecture for DSI, 160–163**
- AreEqual **method, 374**
- AreEquivalent **method, 374**
- AreNotEqual **method, 374**
- AreNotEquivalent **method, 374**
- ASP.NET applications unit testing**
  - attributes of ASP.NET unit tests, 387–388
  - creating ASP.NET unit tests, 388–389
  - overview, 387
- ASP.NET profiling, 336**
- ASP.NETWebApplication, 32**

## **ASP.NETWebService, 32**

### **.ASPX template, 151–153**

### **.ASPX.CS template, 151–153**

#### **Assert methods**

- Assert.AreEqual method, 371–372
- Assert.AreNotEqual method, 371–372
- Assert.AreNotSame method, 372–373
- Assert.AreSame method, 372–373
- Assert.Fail method, 374
- Assert.Inconclusive method, 374
- Assert.IsFalse method, 373
- Assert.IsInstanceOfType method, 373
- Assert.IsNotInstanceOfType method, 373
- Assert.IsNotNull method, 373
- Assert.IsNull method, 373
- Assert.IsTrue method, 373

#### **assigned to field (work items), 517**

#### **assigning roles**

- to activities, 598
- to workstreams, 596

#### **Association item, 103**

#### **association relationships, adding, 112–113**

#### **Attempt to destroy process heap error, 269**

#### **attributes of ASP.NET unit tests, 387–388**

#### **automated deployment, 95–97**

#### **automated testing**

- for generic testing, 435
- for load testing, 435
- manual testing compared, 433–436
- overview, 434–435
- for performance testing, 435
- for regression testing, 435
- for stress testing, 435
- for structural testing, 435
- for web testing, 435

## **B**

#### **backup and recovery, 658–659**

#### **base rule creation, 238**

#### **BigBuild. See Team Foundation Build**

#### **binding applications to servers, 85–87**

#### **BizTalkWebService, 32**

#### **black box testing, 358**

#### **blank performance session, adding, 312**

#### **blogs**

- blogs (Anderson), 570
- blogs (Caron), 648
- blogs (de Vries), 608
- blogs (Harry), 647
- blogs (Hodge), 561
- blogs (Lichten), 632
- blogs (Manning), 608
- blogs (Osterman), 273

#### **branching**

- Team Foundation version control, 553–555
- version control Team Foundation Server, 485

#### **broad partner ecosystem, 159**

#### **Browser Mix (New Load Test Wizard), 415–416**

#### **bug work item**

- logging your manual test as a, 451–452
- overview, 579–580

#### **“Build a Better Design Tool with Visio Automation” (Loton), 22**

#### **build lab, setting up, 629**

#### **build location folder, 629**

#### **build process code analysis integration, 236–237**

#### **build types creation**

- deleting a build type, 635
- editing an existing Team build type, 633–635
- New Team Build Type Creation Wizard used for, 631–633
- overview, 630–631

#### **business analysts, 571, 586–587**

## **C**

#### **Call Tree View (session reports), 327**

#### **Caller/Callee View (session reports), 326–327**

#### **CAP (Call Attributed Provider) tool, 308**

#### **capability levels, 570**

#### **Capability Maturity Model Integration (CMMI)**

- CMMI Level 3, 571–573
- MSF for CMMI Process Improvement architects, 571
- business analysts, 571

- capability levels, 570
- developers, 571
- Lower Control Limits (LCL), 569
- MSF for Agile Software Development compared, 574
- overview, 569–570
- process template, 482
- project managers, 571
- release managers, 571
- testers, 571
- Upper Control Limits (UCL), 569
- capacity planning**
  - enterprise deployment, 651
  - overview, 650
  - performance and scope, 650
  - small to medium deployments, 650
- Caron, Rob (blog), 648**
- C/C++ code analysis**
  - annotation support
    - in C, 254–255
    - in C++, 254
  - overview, 253–254
  - SA\_FormatString attribute, 254
  - SA\_InvalidCheck attribute, 254
  - SA\_No attribute, 254
  - SA\_NoAccess attribute, 254
  - SA\_Post attribute, 254
  - SA\_Pre attribute, 254
  - SA\_Read attribute, 254
  - SA\_ReadWrite attribute, 254
  - SA\_Write attribute, 254
  - SA\_Yes attribute, 254
- code snippets, 300–303
- command-line support, 251–253
- disadvantages of, 262
- Dynamic Link Libraries, 247
- enabling, 248–249
- executables, 247
- how it works, 247–248
- limitations of, 262
- noise, identifying and minimizing, 261–262
- overview, 245–246
- #pragma support, 255–257
- Team Foundation Server integration
  - check-in policies using C/C++ code analysis, creating, 257–259
  - overview, 257
  - setting check-in notes for your C/C++ code, 259–260
  - work items created from code analysis warnings, 260–261
- versions, 246
- Visual Studio 2005 integration
  - code highlighting, 250–251
  - disabling C/C++ code analysis, 248–249
  - enabling C/C++ code analysis, 248–249
  - overview, 248
  - setting warning levels, 249–250
  - viewing code analysis warnings and errors, 250–251
- CCVs (common cause variations), 569**
- CHAOS report (Standish Group), 481**
- check-in and check-outs for Team Foundation version control**
  - checking in an item, 549
  - checking out an item, 550–551
  - policies, creating and administering check-in, 551–553
- check-in policies**
  - using C/C++ code analysis, creating, 257–259
  - version control Team Foundation Server, 486–487
- CI (Continuous Integration), 640–641**
- class, 101–102**
- Class Designer**
  - code synchronization, 120–121
  - described, 16–17
  - designing code types
    - association relationships, adding, 112–113
    - DealingNotificationDelegate delegate, 117
    - DealingUtilities module, 118
    - DealStatus enum, 117
    - hard copy, reproducing class diagrams in, 119
    - inheritance relationships, adding, 110–111
    - interfaces, 118–119

### **Class Designer (continued)**

- members, adding, 111–112
  - methods, 115–116
  - one-to-many associations, 114–115
  - overview, 109–110
  - StockDeal classes, 110–115
  - StockQuote structure, 117
  - types other than class, 116–118
  - dynamic modeling
    - overview, 125
    - simulating collaboration diagrams using comments, 127
    - state pattern used to simulate statechart diagrams, 125–127
  - .NET Framework classes, 121
  - pattern library, building your own, 122–125
  - patterns, 121–122
  - toolbox
    - Abstract Class type, 103
    - Association item, 103
    - Class type, 102
    - Comment tool, 103
    - Delegate type, 103
    - Enum type, 103
    - Inheritance item, 103
    - Interface type, 103
    - Module type, 103
    - overview, 102
    - Structure type, 103
  - UML class diagrams comparison, 128–129
  - visualization of existing code
    - DealingApp class diagram, 104–105
    - members, visualizing, 108–109
    - overview, 103
    - StockBroker example, 104–108
    - StockDeal classes, 105–108
- Class Diagrams, 284–288**
- Class type, 102**
- classification of process templates, 619–621**
- Client zone, 70**
- clients, 490**
- CMMI (Capability Maturity Model Integration)**
- CMMI Level 3, 571–573
  - MSF for CMMI Process Improvement

- architects, 571
- business analysts, 571
- capability levels, 570
- developers, 571
- Lower Control Limits (LCL), 569
- MSF for Agile Software Development compared, 574
- overview, 569–570
- process template, 482
- project managers, 571
- release managers, 571
- testers, 571
- Upper Control Limits (UCL), 569

### **Code Analysis for C/C++**

- annotation support
  - in C, 254–255
  - in C++, 254
  - overview, 253–254
  - SA\_FormatString attribute, 254
  - SA\_InvalidCheck attribute, 254
  - SA\_No attribute, 254
  - SA\_NoAccess attribute, 254
  - SA\_Post attribute, 254
  - SA\_Pre attribute, 254
  - SA\_Read attribute, 254
  - SA\_ReadWrite attribute, 254
  - SA\_Write attribute, 254
  - SA\_Yes attribute, 254
- command-line support, 251–253
- disadvantages of, 262
- Dynamic Link Libraries, 247
- executables, 247
- how it works, 247–248
- limitations of, 262
- noise, identifying and minimizing, 261–262
- overview, 245–246
- #pragma support, 255–257
- Team Foundation Server integration
  - check-in policies using C/C++ code analysis, creating, 257–259
  - overview, 257
  - setting check-in notes for your C/C++ code, 259–260
  - work items created from code analysis warnings, 260–261

- versions, 246
- Visual Studio 2005 integration
  - code highlighting, 250–251
  - disabling C/C++ code analysis, 248–249
  - enabling C/C++ code analysis, 248–249
  - overview, 248
  - setting warning levels, 249–250
  - viewing code analysis warnings and errors, 250–251
- code analysis rules creation**
  - introspection, 237
  - late binding, 237
  - new rules creation
    - base rule creation, 238
    - deploying a rule, 243
    - existing rules, learning from, 243–244
    - IIntrospectionRule interface, 240
    - implementing the rule, 238–243
    - Microsoft.Cci assembly, 239–240
    - overview, 237–238
    - Rules.xml, 242–243
    - writing the implementation code, 240–241
  - overview, 237
  - reflection, 237
- code coverage**
  - and instrumentation, 335
  - overview, 389–390
  - unit testing
    - enabling, 390
    - overview, 389–390
    - results, viewing, 390–391
- code generation**
  - DSL tools, 151–154
  - overview, 382
  - unit testing
    - generating code from tests, 386–387
    - generating tests from code, 382–386
    - overview, 382
- code highlighting, 250–251**
- code refactoring**
  - Encapsulate Field, 291–294
  - Extract Method, 294–296
  - Generate Method Stub, 299–300
  - Promote Local Variable to Parameter, 296–297
  - Remove Parameters, 298–299
  - Rename, 288–291
  - Reorder Parameters, 297–298
- code snippets**
  - Code Snippets Manager, 303–305
  - overview, 300
  - in Visual Basic, 300–302
  - in Visual C++, 302–303
- Code Snippets Manager, 303–305**
- code synchronization**
  - Class Designer, 120–121
  - described, 17–18
- coded web tests, 409–411**
- CollectionAssert **class**
  - AllItemsAreInstancesOfType method, 374
  - AllItemsAreNotNull method, 374
  - AllItemsAreUnique method, 374
  - AreEqual method, 374
  - AreEquivalent method, 374
  - AreNotEqual method, 374
  - AreNotEquivalent method, 374
  - Assert methods, 371–374
  - CollectionAssert class, 374–375
  - Contains method, 374
  - DoesNotContain method, 374
  - IsNotSubsetOf method, 374
  - IsSubsetOf method, 374
  - overview, 374–375
- columns displayed in Test Manager, choosing, 345–346**
- command-line analysis tool**
  - build process code analysis integration, 236–237
  - options, 233–235
  - overview, 233
  - project files, 235–236
- Command-line arguments to pass to the generic test option, 460**
- command-line interface for Team Foundation Build**
  - deleting a build, 638–639
  - nightly builds, setting up, 639–640

### **command-line interface for Team Foundation Build (continued)**

- overview, 638
- starting a build, 638
- stopping a build, 639

### **command-line load test execution, 426**

### **command-line support, 251–253**

### **command-line test execution**

- executing test lists, 426
- executing tests, 426
- options, 427
- overview, 426

### **command-line tools**

- profiling
  - environment variables, setting, 331–332
  - executing a profiling session, 332
  - instrumentation configuration, 330–331
  - overview, 330
  - profiling a Windows service or ASP.NET application, 331
  - reports, generating, 332–333
  - VSInstr.exe utility, 330–331
  - VSPerfClrEnv.cmd utility, 331–332
  - VSPerfCmd.exe utility, 332
  - VSPerfReport.exe utility, 332–333
- Team Foundation version control, 560–561

### **Comment tool**

- Application Designer, 31
- Class Designer, 103
- Logical Datacenter Designer, 60
- System Designer, 82

### **common cause variations (CCVs), 569**

### **Common Object Request Broker Architecture (CORBA), 11, 40**

### **communication between endpoints, 200–201**

### **communication problems and creating a software project, 501**

- CommunicationDefinition relationship, 176

### **complex types, 167–168**

### **component implementation, 11**

### **component interface, 11**

### **components and deploy-time reuse, 9**

### **conceptual overview of Microsoft Solutions Framework (MSF), 565–566**

### **configuration**

- Application Verifier, 264–266
- controllers, 428
- generic testing
  - Additional files to be deployed with this generic test option, 460
  - Command-line arguments to pass to the generic test option, 460
  - Display execution application (window or console) at runtime option, 460
  - Environment Variables option, 460
  - Exit test run if run duration (in milliseconds) exceeds option, 461
  - Redirect standard output and error to test result option, 460
  - Specify an existing program to wrap as a generic test option, 459
- manual test options, 440–442
- Team Foundation architecture, 496–497
- web test run settings, 399

### **configuring agents for distributed load tests**

- attributes, 429
- IP switching, 429
- overview, 428
- weighting, 429

### **Connection tool**

- Application Designer, 31
- Logical Datacenter Designer, 60
- System Designer, 82

### **Connector Map, providing a, 145–146**

### **connectors, defining, 143**

### **Conserve Memory setting (Application Verifier), 264**

### **console applications, 56**

- ConstraintDefinition complex type, 182–183

- ConstraintMember complex type, 178, 183–184

### **constraints**

- ConstraintDefinition complex type, 182–183
- ConstraintMember complex type, 183–184

- creating your own, 211–216
- described, 71
- enforcing, 215–216
- Evaluate attribute, 183
- FlowMember complex type, 184–185
- ManagerDeclaration complex type, 185–186
- overview, 182
- RaiseError attribute, 183
- RelationshipConstraints, 208–209
- ReturnEarly attribute, 183
- SDM, 162, 208–216
- SdmG (SDM Manager Generator), 212–215
- setting constraints, 182
- settings compared, 71–72
- structural constraints, 182
- System Definition Model (SDM), 182–186
- TargetDefinition attribute, 183
- validation, 209–211
- consumer endpoints, 31**
- containment, 201–205**
- ContainmentDefinition relationship, 176
- Contains method, 374, 375
- context parameters, 404–405**
- Continuous Integration (CI), 640–641**
- contributor role, 543**
- Cooper, Alan (*The Inmates are Running the Asylum*), 576**
- copying diagrams, 54**
- CORBA (Common Object Request Broker Architecture), 11, 40**
- core services, 489**
- core template, 605–607**
- correcting problems with rule violations, 230–232**
- Corrupted end stamp of block header error, 270**
- Corrupted infix pattern for freed block error, 269**
- Corrupted prefix pattern error, 270**
- Corrupted start stamp of block header error, 270**
- Corrupted suffix pattern error, 269**
- cosmetic bugs, 438**
- Counter Sets (New Load Test Wizard), 416–417**
- Counters property page (Performance Explorer), 315**
- creating**
  - ASP.NET unit tests, 388–389
  - custom manual test templates, 443–445
  - generic tests, 456–462
  - load tests, 411–418
  - manual tests, 439
  - new activities, 597–598
  - new workstreams, 596
  - ordered tests, 352–353
  - software project
    - communication problems, 501
    - failure of software development projects, reasons for, 501–502
    - metrics and estimation skills, lack of, 501
    - New Team Project Wizard, 502–505
    - operational details, too much focus on, 501
    - overview, 500–501
    - poor workflow management, 501
    - standards compliance, lack of, 501
    - Team Project, 500
    - understanding of processes, lack of, 501
    - Visual Studio Project, 500
  - system diagrams from scratch, 92
  - test lists, 344–345
  - a test plan, 654
  - test projects, 340–341
  - Visual Studio 2005 test template, 446
  - web tests, 396–397
  - your first unit test
    - failing tests, 363
    - identifying unit test classes, 363
    - identifying unit tests, 363
    - overview, 362–363
    - successful tests, 363
    - your own constraints, 211–216
- critical bugs, 438**
- Critical section over-released or corrupted error, 272**
- cross-references, 600**

## **Cunningham, Ward (Framework for Integrated Testing), 434**

**The current thread is performing a wait operation on a thread handle in DllMain error, 271**

### **custom reports**

- building, 530–532
- overview, 530–531
- Report Builder used to create, 531–532
- Report Definition Language files, 530
- running, 532
- viewing, 532

**custom roles, creating, 600**

**custom settings, 75, 181–182**

**custom unit test properties, defining, 377**

### **customization**

- manual test added to your project, 443
- process templates
  - definitions, modifying template, 603–622
  - downloading a process template to your desktop, 601–602
  - navigating a process template, 602–603
  - overview, 601
- Team Foundation Build, 644–646
- work item elements, 599

### **customization tools**

- Microsoft InfoPath 2003, 593–594
- MSF HTML Build utility, 594
- overview, 592–593
- third-party tools, 594–595
- XML/XSL editors, 593

**cycles in Microsoft Solutions Framework (MSF), 575–576**

## **D**

**database connection properties, specifying, 36–37**

**DatabaseClientEndpoint (Logical Datacenter Designer), 61**

**DatabaseServer (Logical Datacenter Designer), 61**

**datacenter zones, 63**

**data-driven unit tests, creating, 378**

**data-driven web tests**

- binding to a source, 408–409

- data source configuration, 407–408
- overview, 407

**DCOM (Distributed Component Object Model) applications, 11, 55**

**de Vries, Marcel (blog), 608**

**DealingApp class diagram, 104–105**

**DealingApp implementation**

- DealingForm.vb code, 51
- overview, 50–51
- testing DealingApp Windows application, 51–52

**DealingForm.vb code, 51**

**DealingNotificationDelegate delegate, 117**

**DealingService.vb, 46–47**

**DealingUtilities module, 118**

**DealStatus enum, 117**

***Debugging Applications for Microsoft .NET and Microsoft Windows (Robbins), 266***

**debugging symbols, 335**

**debugging unit tests, 368**

**default deployments, 88–89**

**default templates, 442–443**

**Default.aspx file, 49**

**Default.aspx.cs file, 49–50**

**defining and deploying systems**

- defining deployment, 90–91
- nested systems, 91–92
- overview, 89
- system definition for a subset of the application design, 89–90
- system definition for the full application design, 89

**defining default deployment from Application Designer**

- binding applications to servers, 85–87
- overview, 83–85
- prohibited bindings, 87
- System View, 85
- validating the deployment, 87–88

**defining deployment, 90–91**

**Definition complex type, 170–171**

**definition members, 170**

**definitions, modifying process template**

- classification, 619–621
- core template, 605–607

- groups, 621–622
- overview, 603–605
- permissions, 621–622
- SQL Reporting Services, 617–619
- Team Foundation version control, 607–609
- testing your changes, 613
- Windows SharePoint Services team portal, 613–617
- work items, 609–612
- Delegate type, 103**
- Delegation of endpoints, 200–201**
- DelegationDefinition relationship, 176
- deleting**
  - a build, 638–639
  - a build type, 635
  - process templates, 623
  - a Team project, 514–516
- Delrue, Thomas (SDM example), 192**
- Deming, W. Edwards (statistician), 569**
- deploying a rule, 243**
- deployment, 165–166**
- Deployment Designer**
  - deployment reports
    - automated deployment, 95–97
    - human-readable deployment reports, 94–95
    - machine-readable deployment reports, 95–97
    - overview, 93–94
  - overview, 15–16, 82
  - System View, 83
  - UML deployment diagram compared, 97–98
- deployment planning**
  - capacity planning
    - enterprise deployment, 651
    - overview, 650
    - performance and scope, 650
    - small to medium deployments, 650
  - creating a test plan, 654
  - firewall configuration, 652
  - network topologies
    - architecting your Active Directory (AD) structure, 653–654
    - multi-server deployment, 652
    - single-server deployment, 652
    - test deployment using virtualization, 654
  - overview, 649
  - test rig considerations, 655
  - unsupported deployment scenarios, 650
  - user accounts, 651
- deployment reports**
  - automated deployment, 95–97
  - human-readable deployment reports, 94–95
  - machine-readable deployment reports, 95–97
  - overview, 93–94
- DeploymentTime attribute, 179
- design characteristics, 33–34**
- “Design Guidelines for Class Library Developers” (Microsoft), 224**
- design rules, 225**
- designer, creating a, 132–133**
- designer relationships and team system integration**
  - object flow (artifacts), 20–21
  - overview, 19
  - process flow (roles and activities), 19–20
  - settings and constraints, 21
  - source control and item tracking, 21
- designing code types**
  - association relationships, adding, 112–113
  - DealingNotificationDelegate delegate, 117
  - DealingUtilities module, 118
  - DealStatus enum, 117
  - hard copy, reproducing class diagrams in, 119
  - inheritance relationships, adding, 110–111
  - interfaces, 118–119
  - members, adding, 111–112
  - methods, 115–116
  - one-to-many associations, 114–115
  - overview, 109–110
  - StockDeal classes, 110–115
  - StockQuote structure, 117
  - types other than class, 116–118
- designing visually, reasons for, 4**
- design-time validation, 163–164**
- developers, 571, 588**
- development role, 543**
- disabling C/C++ code analysis, 248–249**
- disabling tests, 344**

**Display execution application (window or console) at runtime option (generic testing), 460**

**Distributed Component Object Model (DCOM), 11, 55**

**distributed components and run-time reuse, 10–11**

**distributed load tests**

configuring agents

attributes, 429

IP switching, 429

overview, 428

weighting, 429

load tests

configuring agents, 428–429

configuring controllers, 428

installing controllers and agents, 427

overview, 427

running, 430

test run configuration, 429–430

viewing, 430

overview, 427

**distributed services and service-oriented architecture, 11–12**

**Distributed System Designers, 189–190**

**DMZ1 zone, 63**

**DMZ2 zone, 63**

**DoesNotContain method, 374**

**DoesNotMatch method, 375**

**domain models, 131, 149–150**

**domain-specific language (DSL), 5–7, 131–132**

**downloading**

Microsoft Solutions Framework (MSF), 566

a process template to your desktop, 601–602

**drawing the diagram, 34–36**

**drop location folder, 629**

**DSI. See Dynamic Systems Initiative**

**DSI roadmap, 159**

**DSL toolkit, 25**

**DSL Tools**

.ASPX template, 151–153

.ASPX.CS template, 151–153

code generation, 151–154

designer, creating a, 132–133

domain model, using, 149–150

graphical language, creation of

Connector Map, providing a, 145–146

connectors, defining, 143

overview, 141–142

Shape Map, providing a, 144–145

shapes, defining, 142–143

toolbox, defining, 144

latest release of, 132

navigation language, creation of, 134–141

pages, 135–137

project, creation of, 133–134

relationships, 137–138

sites, 135–137

translation code, 155

using the designer, 147–149

**Dynamic Link Libraries, 247**

**dynamic modeling**

Application Designer, 56–57

Class Designer

overview, 125

simulating collaboration diagrams using comments, 127

state pattern used to simulate statechart diagrams, 125–127

limitations and workarounds, 56–57

System Designer, 98

**Dynamic Systems Initiative (DSI)**

broad partner ecosystem, 159

DSI roadmap, 159

overview, 159

System Definition Model (SDM)

abstract attribute, 172

abstract objects, 161

abstract relationships, 161

advantages of, 160

architecture for, 160–163

complex types, 167–168

constraints, 162, 182–186

Definition complex type, 170–171

definition members, 170

deployment, 165–166

design-time validation, 163–164

Distributed System Designers, 189–190

example, 192–218  
 extends attribute, 172  
 instances, 161–162  
 layer attribute, 172  
 managers, 162  
 Member complex type, 176–179  
 modeling layers of a system, 166–167  
 object definition, 170  
 ObjectDefinition complex type,  
     172–175  
 overview, 160  
 prototypes, 189–190  
 relationship definition, 170  
 RelationshipDefinition complex  
     type, 175–176  
 schema structure, 169–170  
 SDM Compiler, 163, 191–192  
 SDM documents, 162, 186–189  
 SDM SDK, 190–191  
 settings, 179–182  
 simple types, 167–168  
 system deployment definition, 166  
 type definitions, 168  
 types, 167–169  
 in Visual Studio 2005 Team Edition for Soft-  
     ware Architects, 164–165

## E

### editing

an existing Team build type, 633–635  
 load tests, 419–422  
 web tests  
     comments, 403–404  
     overview, 403  
     request properties, 403  
     transactions, 404

### **EJB (Enterprise Java Beans), 11**

#### **elapsed time, 322**

#### **e-mail notifications, 635–636**

#### **Encapsulate Field, 291–294**

#### **endpoint styles, 64**

#### **endpoints**

Application Designer, 31  
 communication between, 200–201

delegation of, 200–201  
 details, specifying, 37–38  
 Logical Datacenter Designer, 61  
 prototype, defining your own, 38–39  
 SDM, 196–200

#### **EndsWith method, 375**

#### **enforcing constraints, 215–216**

#### **enterprise deployment, 651**

#### **Enterprise IT Administrator, 649**

#### **Enterprise Java Beans (EJB), 11**

#### **entities, 532**

#### **entry and exit criteria in workstreams, 596–597**

#### **Enum type, 103**

#### **Environment Variables option (generic test- ing), 460**

#### **ErrorLevel codes (generic testing), 461–462**

#### **Errors table, 425–426**

#### **estimation skills, 501**

#### **Evaluate attribute, 183**

#### **Events property page (Performance Explorer), 315**

#### **example (SDM)**

Application layer endpoint, 196–198  
 ApplicationHost layer endpoint, 199–200  
 communication, 200–201  
 constraints  
     creating your own, 211–216  
     enforcing, 215–216  
     RelationshipConstraints, 208–209  
     SdmG (SDM Manager Generator), 212–215  
     validation, 209–211  
 containment, 201–205  
 delegation, 200–201  
 endpoints, 196–200  
 flows, 207  
 hosting, 206  
 ProtoGen, 216–217  
 registry, 217–218  
 SDM Compiler, 216  
 settings, 195–196  
 settings for endpoints, 198–199  
 Visual Studio and, 218–219  
**example summary report XML file, 469–470**

- Exception raised while verifying block header error, 269**
- exceptions, 376**
- Excessive size for the current operation error, 268**
- exclusive time, 322**
- executables, 247**
- executing**
  - load tests, 422
  - ordered tests, 354
  - test lists, 426
  - web tests, 402–403
- existing models and UML, 24**
- existing rules, learning from, 243–244**
- existing service, representing an, 39–40**
- exit codes, 461–462**
- Exit test run if run duration (in milliseconds) exceeds option (generic testing), 461**
- `ExpectedException` attribute, 376
- exporting test results, 350**
- exporting your custom manual test as a new item, 443–445**
- extended return results**
  - example summary report XML file, 469–470
  - overview, 467–468
  - summary report XML schema, 468–469
- extends attribute, 172**
- extensibility, 644–646**
- external testing tool creation for generic testing**
  - Managed Code, 465–467
  - overview, 464
  - reasons for, 464
  - Windows Scripting Host, 464–465
- ExternalDatabase (Application Designer), 32**
- ExternalWebService (Application Designer), 32**
- Extract Interface, 284–287**
- Extract Method, 294–296**
- extraction rules, 404–405**

## F

- failure of software development projects, reasons for, 501–502**

- filtering test results, 350**
- filtering tests, 346–347**
- firewall configuration, 652**
- `FlowMember` complex type, 178, 184–185
- flows, 207**
- Foemmel, Matthew (CI), 641**
- Fowler, Martin (CI), 641**
- Framework for Integrated Testing (FIT) (Cunningham), 434**
- Full page heap, 267**
- fully integrated UML capability, achieving, 24–25**
- Functions View for session reports, 325–326**
- FxCop, 224**
- FxCopCmd**
  - build process code analysis integration, 236–237
  - options, 233–235
  - overview, 233
  - project files, 235–236

## G

### General Designer

- Application Designer, 31
- Logical Datacenter Designer, 60

### General property page (Performance Explorer), 314–315

- general session properties, setting, 314–316**

### Generate Method Stub, 299–300

- generating code from tests, 386–387**

- generating tests from code, 382–386**

### generic testing

- automated testing, 435
- configuration options
  - Additional files to be deployed with this generic test option, 460
  - Command-line arguments to pass to the generic test option, 460
  - Display execution application (window or console) at runtime option, 460
  - Environment Variables option, 460
  - Exit test run if run duration (in milliseconds) exceeds option, 461

- Redirect standard output and error to test result option, 460
  - Specify an existing program to wrap as a generic test option, 459
  - creating a generic test, 456–462
  - ErrorLevel codes, 461–462
  - exit codes, 461–462
  - extended return results
    - example summary report XML file, 469–470
    - overview, 467–468
    - summary report XML schema, 468–469
  - external testing tool creation
    - Managed Code, 465–467
    - overview, 464
    - reasons for, 464
    - Windows Scripting Host, 464–465
  - overview, 455
  - running the generic test, 462–463
  - test projects, 340
  - wiring to existing test systems
    - JScript conversion code, 473–477
    - NUnit, creating a wire to, 471–473
    - overview, 470–471
  - GenericApplication (Application Designer), 32**
  - GenericClientEndpoint (Logical Datacenter Designer), 61**
  - GenericEndpoint (Application Designer), 31**
  - GenericServer (Logical Datacenter Designer), 61**
  - GenericServer settings and constraints, 72**
  - GenericServerEndpoint (Logical Datacenter Designer), 61**
  - globalization rules, 225**
  - goal-based load profiles, 420–421**
  - Goldratt, Eliyahu M. (*The Goal*), 569**
  - Goodwin, Kim (personas), 576**
  - governance, 577**
  - graphical language creation**
    - Connector Map, providing a, 145–146
    - connectors, defining, 143
    - overview, 141–142
    - Shape Map, providing a, 144–145
    - shapes, defining, 142–143
    - toolbox, defining, 144
  - Graphs View of results, 423–424**
  - Greenfield, Jack (*Software Factories: Assembling Applications with Patterns, Works, Models and Tools*), 5, 7**
  - grouping test results, 350**
  - grouping tests, 346**
  - groups, security, 621–622**
  - Guthrie, Scott (*MadDog*), 437**
- ## H
- handle troubleshooting**
    - The current thread is performing a wait operation on a thread handle in DIIMain error, 271
    - The handle used for the current operation is invalid error, 270
    - Incorrect object type for handle error, 271
    - Invalid parameters passed to a multi-object wait function error, 270
    - Invalid TLS index in the current operation error, 271
    - A NULL handle has been used error, 271
    - overview, 270
  - The handle used for the current operation is invalid error, 270**
  - hard copy, reproducing class diagrams in, 119**
  - Harry, Brian (blog), 647**
  - Heap handle with incorrect signature error, 268**
  - Heap operation performed on an invalid heap handle error, 268**
  - heap troubleshooting**
    - Attempt to destroy process heap error, 269
    - Corrupted end stamp of block header error, 270
    - Corrupted infix pattern for freed block error, 269
    - Corrupted prefix pattern error, 270
    - Corrupted start stamp of block header error, 270
    - Corrupted suffix pattern error, 269
    - Exception raised while verifying block header error, 269
    - Excessive size for the current operation error, 268

### heap troubleshooting (continued)

- Full page heap, 267
- The heap block object of the current operation is corrupted error, 268–269
- Heap handle with incorrect signature error, 268
- Heap operation performed on an invalid heap handle error, 268
- Light page heap, 267
- Memory Access Operation in the context of a freed block reuse-after-delete or double-delete error, 267
- Memory Access Operation in the context of an allocated block: heap overrun or heap underrun error, 267
- Multithreaded access in a `HEAP_NO_SERIALIZE` heap error, 267–268
- overview, 266–267
- Process heap list count is wrong error, 270
- Unexpected exception raised in heap code path error, 269

**The heap block object of the current operation is corrupted error, 268–269**

**history, 485**

**Hodge, Buck (blog), 561**

**hosting, 206**

`HostingDefinition` relationship, 176

**hotspots, 319**

**HTTPClientEndpoint (Logical Datacenter Designer), 61**

**human-readable deployment reports, 94–95**

**Hunter, Michael (testing methodologies), 453**

## I

**ID field (work items), 516**

`IInspectionRule` interface, 240

**IISWebServer**

overview, 61

settings and constraints, 72–74

**Implement Abstract Class, 287–288**

**implementation, component, 11**

**implementation of applications**

Application Designer

DealingApp implementation, 50–52

MarketMaker implementation, 43–45

overview, 41–42

SDM files, 42–43

StockBroker implementation, 45–48

StockQuoteApp implementation, 48–50

DealingApp implementation

DealingForm.vb code, 51

overview, 50–51

testing DealingApp Windows application, 51–52

MarketMaker implementation

modifications to DealingService.cs, 43–44

overview, 43

testing MarketMaker application, 44–45

StockBroker implementation

DealingService.vb, 46–47

overview, 45

StockQuoteService.vb, 46

testing StockBroker application, 47–48

StockQuoteApp implementation

Default.aspx file, 49

Default.aspx.cs file, 49–50

overview, 48

testing StockQuoteApp Web application, 50

**implementing the rule, 238–243**

**Import IIS Settings Wizard, 76**

**inclusive time, 322**

**Incorrect object type for handle error, 271**

**Inheritance item, 103**

**inheritance relationships, adding, 110–111**

**initialization and cleanup of unit tests**

AssemblyCleanup attribute, 371

AssemblyInitialize attribute, 371

ClassCleanup attribute, 370–371

ClassInitialize attribute, 370–371

overview, 368–369

TestCleanup attribute, 369–370

TestInitialize attribute, 369–370

***The Inmates are Running the Asylum* (Cooper), 576**

**installing controllers and agents, 427**

**instances, 161–162**

**instrumentation**

and code coverage, 335

programmatically modifying, 333–335

**instrumentation profilers, 308**  
**instrumentation session configuration, 319–320**  
**integrated check-in, 484**  
**integrating your custom manual test template into Visual Studio 2005, 446–447**  
**integration**  
 CI (Continuous Integration), 640–641  
 overview, 656  
 reporting, 657  
 Team Foundation Server integration  
   check-in policies using C/C++ code analysis, creating, 257–259  
   overview, 257  
   setting check-in notes for your C/C++ code, 259–260  
   work items created from code analysis warnings, 260–261  
 testing tools, 657–658  
 version control, 656  
 Visual Studio 2005 integration  
   code highlighting, 250–251  
   disabling C/C++ code analysis, 248–249  
   enabling C/C++ code analysis, 248–249  
   overview, 248  
   setting warning levels, 249–250  
   viewing code analysis warnings and errors, 250–251  
   work-item tracking, 657  
**Interface type, 103**  
**interfaces, 118–119**  
**interoperability rules, 225**  
**interpreting session reports, 321–330**  
**IntranetZone, 63**  
**introspection, 237**  
**Invalid parameters passed to a multi-object wait function error, 270**  
**Invalid TLS index in the current operation error, 271**  
 IsNotSubsetOf **method, 374**  
 IsSubsetOf **method, 374**  
**iterative process, 575**

## J

**JScript conversion code, 473–477**

## L

**late binding, 237**

**Launch property page (Performance Explorer), 315**

**layer attribute, 172**

**LCL (Lower Control Limits), 569**

**licensing models, 659**

**Lichten, Craig (blog), 632**

**Light page heap, 267**

**limitations and workarounds**

  Application Designer

    console applications, 56

    DCOM applications, 55

    dynamic modeling, 56–57

    .NET Remoting, 55

    overview, 54

    Windows applications and services, 56

  overview, 54

**limitations of Code Analysis for C/C++, 262**

**load patterns, 412–413**

**Load Test Editor**

  goal-based load profiles, 420–421

  overview, 419

  run settings, 420

  scenarios, adding, 419–420

  SQL Tracing, 420

  storing load test run data, 422

**load testing**

  automated testing, 435

  command-line test execution

    executing test lists, 426

    executing tests, 426

    options, 427

    overview, 426

  creating and configuring, 411–418

  distributed load tests

    configuring agents, 428–429

    configuring controllers, 428

### load testing (continued)

- installing controllers and agents, 427
- overview, 427
- running, 430
- test run configuration, 429–430
- viewing, 430
- editing, 419–422
- executing, 422
- Load Test Editor
  - goal-based load profiles, 420–421
  - overview, 419
  - run settings, 420
  - scenarios, adding, 419–420
  - SQL Tracing, 420
  - storing load test run data, 422
- New Load Test Wizard
  - Browser Mix, 415–416
  - Counter Sets, 416–417
  - load patterns, 412–413
  - Network Mix, 416
  - New Load Test Wizard, 411–413
  - overview, 411
  - performance counters, 416–417
  - Run Settings, 417–418
  - scenarios, 411–412
  - Test Mix, 413–415
  - think times, 412
- overview, 411
- results
  - Errors table, 425–426
  - Graphs View of, 423–424
  - overview, 422–423
  - Pages table, 424
  - SQL Trace table, 425
  - Tables View of, 424–426
  - Tests table, 424
  - Thresholds table, 425
  - Transactions table, 424
- test projects, 340
- local build, executing, 636–637**
- lock troubleshooting**
  - Critical section over-released or corrupted error, 272
  - The lock count is invalid error, 273

- The Lock has already been initialized error, 272
- The Lock has been double initialized error, 272
- The lock has been over released error, 273
- The lock is being used without having been initialized error, 273
- The lock is corrupted error, 272–273
- Module contains an active lock error, 272
- overview, 271–272
- The owner of the lock is invalid, owner was expected error, 273
- The recursion count is invalid error, 273
- The region of memory about to be freed contains an active lock error, 273
- Released memory containing an active lock error, 272
- Released virtual memory containing an active lock error, 272
- The thread cannot own a lock in this context error, 272
- Thread not supposed to be owning a lock error, 273
- The lock count is invalid error, 273**
- The Lock has already been initialized error, 272**
- The Lock has been double initialized error, 272**
- The lock has been over released error, 273**
- The lock is being used without having been initialized error, 273**
- The lock is corrupted error, 272–273**
- Logical Datacenter Designer**
  - overview, 60
  - reusable prototypes
    - overview, 77
    - sharing, 78
    - StockBrokerZone prototype, 77–78
  - sample logical datacenter
    - datacenter zones, 63
    - DMZ1 zone, 63
    - DMZ2 zone, 63
    - endpoint styles, 64
    - IntranetZone, 63

- overview, 62–63
  - UML deployment diagram, 64–65
  - settings and constraints
    - constraints versus settings, 71–72
    - custom settings, 75
    - GenericServer settings and constraints, 72
    - IISWebServer settings and constraints, 72–74
    - Import IIS Settings Wizard, 76
      - overview, 71
      - user-defined constraints, 76
    - WebSiteEndpoint settings and constraints, 74
    - ZoneEndpoint settings and constraints, 74–75
  - Stockbroker example
    - Client zone, 70
    - MarketMaker zone, 65–67
      - overview, 65
    - StockBroker zone, 68–69
    - StockMarket zone, 67–68
  - toolbox
    - Comment tool, 60
    - Connection tool, 60
    - DatabaseClientEndpoint, 61
    - DatabaseServer, 61
    - endpoints, 61
    - General Designer, 60
    - GenericClientEndpoint, 61
    - GenericServer, 61
    - GenericServerEndpoint, 61
    - HTTPClientEndpoint, 61
    - IISWebServer, 61
    - Logical Servers, 61
      - overview, 60
    - WebSiteEndpoint, 61
    - WindowsClient, 61
    - Zone, 61
    - ZoneEndpoint, 61
  - Logical Servers (Logical Datacenter Designer), 61**
  - Loton, Tony (“Build a Better Design Tool with Visio Automation”), 22**
  - Lower Control Limits (LCL), 569**
  - Low-Overhead Profiler (LOP) tool, 308**
  - Lucovsky, Jeff (project alerts), 513**
- ## M
- machine-readable deployment reports, 95–97**
  - MadDog (Guthrie), 437**
  - maintainability rules, 225**
  - Managed Code, 465–467**
  - Managed Code Analysis**
    - design rules, 225
    - enabling, 226–227
    - globalization rules, 225
    - interoperability rules, 225
    - maintainability rules, 225
    - mobility rules, 225
    - naming rules, 226
    - overview, 224–225
    - performance rules, 226
    - portability rules, 226
    - reliability rules, 226
    - rule violations
      - correcting problems, 230–232
      - overview, 230
      - suppressing messages, 232–233
    - rules for, built-in, 225–226
    - running, 227–229
    - security rules, 226
    - usage rules, 226
  - ManagerDeclaration complex type, 185–186**
  - managers, 162**
  - managing**
    - manual tests, 449–453
    - session reports, 321
  - managing and running unit tests**
    - overview, 364
    - Test Manager, 365–366
    - Test View, 365
  - Manifesto for Agile Software Development, 568**
  - Manning, James (blog), 608**
  - manual testing**
    - automated testing compared, 433–436
    - cosmetic bugs, 438
    - critical bugs, 438

### **manual testing (continued)**

- medium bugs, 438
- minor bugs, 438
- overview, 433, 435–436
- planning for, 437–438
- prioritization scale, 438
- severe bugs, 438
- in Team System
  - bug work item, logging your manual test as a, 451–452
  - configuring manual test options, 440–442
  - creating manual tests, 439
  - managing manual tests, 449–453
  - overview, 438–439
  - structure, 448–449
  - Team Foundation Server, publishing test results on, 452–453
  - templates, 442–448
  - Test Manager, 450–451
- test cases, 437
- test projects, 340

**MarketMaker implementation**

- modifications to `DealingService.cs`, 43–44
- overview, 43
- testing MarketMaker application, 44–45

**MarketMaker zone, 65–67**

**MASM (Microsoft Assembler Macro Language), 276**

`Matches` **method, 375**

`MaxOccurs` **attribute, 178**

**MDA (Model-Driven Architecture), 5**

**medium bugs, 438**

`Member` **complex type, 176–179**

**members**

- adding, 111–112
- visualizing, 108–109

**Memory Access Operation in the context of a freed block reuse-after-delete or double-delete error, 267**

**Memory Access Operation in the context of an allocated block: heap overrun or heap underrun error, 267**

**merging**

- Team Foundation version control, 555–557
- version control Team Foundation Server, 485

**metadata file, 341**

**meta-models, 22**

**methods, 115–116**

**metrics and estimation skills, 501**

**Microsoft Assembler Macro Language (MASM), 276**

**Microsoft (“Design Guidelines for Class Library Developers”), 224**

**Microsoft Excel**

- pivot tables, 529–530
- used to create and assign work items, 519–521

**Microsoft InfoPath 2003, 593–594**

**Microsoft Office used to create and assign work items, 519–522**

**Microsoft Project used to create and assign work items, 521–522**

**Microsoft Solutions Framework (MSF)**

- activities, 578
- conceptual overview, 565–566
- cycles, 575–576
- downloading, 566
- governance, 577
- iterative process, 575
- MSF for Agile Software Development
  - Agile movement, 568–569
  - Manifesto for Agile Software Development, 568
  - MSF for CMMI Process Improvement compared, 574
  - overview, 566–567
- MSF for CMMI Process Improvement
  - architects, 571
  - business analysts, 571
  - capability levels, 570
  - CMMI Level 3, 571–573
  - developers, 571
  - Lower Control Limits (LCL), 569
  - MSF for Agile Software Development compared, 574
  - overview, 569–570
  - project managers, 571
  - release managers, 571
  - testers, 571
  - Upper Control Limits (UCL), 569

- overview, 563–566
- personas, 576–577
- process template components
  - overview, 574
  - work item queries, 574–575
  - work products, 574
- reports, 582
- roles
  - architect, 587–588
  - business analyst, 586–587
  - developer, 588
  - overview, 585–586
  - project manager, 587
  - release manager, 589
  - tester, 588–589
  - user experience, 589
- Team Foundation Server, 482
- Team Model
  - advocacy, 584–585
  - mindsets, 585
  - overview, 582–584
- tracks, 577–578
- work item types
  - bug work item, 579–580
  - overview, 578–579
  - quality of service (QOS) requirement work item, 580–581
  - risk work item, 581–582
  - scenario work item, 581
  - task work item, 582
- workstreams, 589–590
- Microsoft Transaction Server (MTS), 11**
- Microsoft Word format template, 447–448**
- `Microsoft.Cci` **assembly, 239–240**
- Microsoft’s modeling strategy**
  - domain-specific languages, 6–7
  - model-driven development, 5–6
  - overview, 5
  - software factories, 7–8
- migration**
  - overview, 655
  - reporting, 657
  - testing tools, 657–658
  - version control, 656
  - from VSS to Team Foundation version control, 487
  - work-item tracking, 657
- mindsets, 585**
- `MinOccurs` **attribute, 178**
- minor bugs, 438**
- mobility rules, 225**
- model and code synchronization, 52**
- Model-Driven Architecture (MDA), 5**
- model-driven development, 5–6**
- modeling layers of a system, 166–167**
- modifications to `DealingService.cs`, 43–44**
- Module contains an active lock error, 272**
- Module type, 103**
- MSBuild tasks, 645–646**
- MSF. See Microsoft Solutions Framework**
- MSF for Agile Software Development**
  - Agile movement, 568–569
  - Manifesto for Agile Software Development, 568
  - MSF for CMMI Process Improvement compared, 574
  - overview, 566–567
  - process template, 482
- MSF for CMMI Process Improvement**
  - architects, 571
  - business analysts, 571
  - capability levels, 570
  - CMMI Level 3, 571–573
  - developers, 571
  - Lower Control Limits (LCL), 569
  - MSF for Agile Software Development compared, 574
  - overview, 569–570
  - process template, 482
  - project managers, 571
  - release managers, 571
  - testers, 571
  - Upper Control Limits (UCL), 569
- MSF HTML Build utility, 594**
- MTS (Microsoft Transaction Server), 11**
- multi-server deployment, 652**
- Multithreaded access in a `HEAP_NO_SERIALIZE` heap error, 267–268**

## N

**naming rules, 226**

**navigating a process template, 602–603**

**navigation language, creation of, 134–141**

**nested systems, 91–92**

**.NET Remoting, 55**

**Network Mix (New Load Test Wizard), 416**

**network topologies**

architecting your Active Directory (AD) structure, 653–654

multi-server deployment, 652

single-server deployment, 652

test deployment using virtualization, 654

**New Load Test Wizard**

Browser Mix, 415–416

Counter Sets, 416–417

load patterns, 412–413

Network Mix, 416

overview, 411

performance counters, 416–417

Run Settings, 417–418

scenarios, 411–412

Test Mix, 413–415

think times, 412

**new rules creation**

base rule creation, 238

deploying a rule, 243

existing rules, learning from, 243–244

`IInspectionRule` interface, 240

implementing the rule, 238–243

`Microsoft.Cci` assembly, 239–240

overview, 237–238

`Rules.xml`, 242–243

writing the implementation code, 240–241

**New Team Build Type Creation Wizard used for Team Foundation Build, 631–633**

**New Team Project Wizard**

creating a software project, 502–505

process templates, 592

**new visual designers, UML comparisons to, 23–24**

**Newkirk, James (NUnit Converter), 361, 657**

**nightly builds, setting up, 639–640**

**noise, identifying and minimizing, 261–262**

**NUnit**

creating a wire to, 471–473

overview, 361

**NUnit Converter (Newkirk), 361, 657**

## O

**object definition, 170**

**object flow (artifacts), 20–21**

**object flow diagram, 19**

**Object Management Group (OMG), 5**

**object-based architectures evolution to service-oriented architectures**

components and deploy-time reuse, 9

distributed components and run-time reuse, 10–11

distributed services and service-oriented architecture, 11–12

objects and compile-time reuse, 8–9

overview, 8

`ObjectDefinition` **complex type, 172–175**

`ObjectMember` **complex type, 178**

**objects and compile-time reuse, 8–9**

**Objects Lifetime View in session reports, 328–330**

**OfficeApplication (Application Designer), 32**

**one-to-many associations, 114–115**

**opening test templates using Open With Wizard, 447**

**operational details, too much focus on, 501**

**operations team, setting up, 648–649**

**options**

command-line analysis tool, 233–235

command-line load test execution, 427

test projects, 341–342

**ordered tests**

analyzing, 354–355

creating, 352–353

executing, 354

overview, 352

properties of, 353

test projects, 340

**Osterman, Larry (blog), 273**

**The owner of the lock is invalid, owner was expected error, 273**

## P

**pages, 135–137**

**Pages table, 424**

**parameterizing the web server, 399–400**

**PBSystems Test Case Manager, 437**

**performance analysis. See also Team Developer profiler**

instrumentation profilers, 308

overview, 307–308

sampling profilers, 308

Team System profiling, 308

tracing markers, 308

types of profilers, 308

**performance and scope of deployments, 650**

**performance counters, 416–417**

**Performance Explorer**

overview, 313–314

Team Developer profiler

Counters property page, 315

Events property page, 315

General property page, 314–315

general session properties, setting,  
314–316

Launch property page, 315

overview, 313–314

session targets, configuring, 316–318

**performance rules, 226**

**performance session creation**

overview, 311

Team Developer profiler

blank performance session, adding, 312

overview, 311

unit test, creating a performance session  
from a, 312–313

using Performance Wizard, 311–312

**performance session execution, 320**

**performance testing, 435**

**permissions, 621–622**

**personas, 576–577**

**planning for manual testing, 437–438**

**platform-specific models (PSMs), 5**

**pointer tool (Application Designer), 31**

**policies, creating and administering check-in,  
551–553**

**poor workflow management, 501**

**populating test lists, 345**

**portability rules, 226**

**#pragma support, 255–257**

**PREsharp, 245**

**printing diagrams, 54**

**prioritization scale for manual testing, 438**

**private members testing**

overview, 379

unit testing

overview, 379

PrivateObject class, 379–381

PrivateType class, 381–382

**process flow diagram, 19**

**process flow (roles and activities), 19–20**

**process guidance, 499**

**process guidance customization**

activities

adding steps to, 598

assigning roles to, 598

creating new activities, 597–598

referencing work products, 597

workstreams, associating activities to, 598

process templates

activities, 597–598

cross-references, 600

custom roles, creating, 600

customizing work item elements, 599

overview, 595

ProcessGuidance.xml, 595–596

viewing your custom guidance, 601

work item documentation, 599

workstreams, 596–597

workstreams and activities, linking work  
items to, 599

workstreams

assigning roles to, 596

creating new workstreams, 596

entry and exit criteria in, 596–597

overview, 596

**Process heap list count is wrong error, 270**

### **process template components**

- Microsoft Solutions Framework (MSF)
  - overview, 574
  - work item queries, 574–575
  - work products, 574
- overview, 574

### **process templates**

- customization
  - definitions, modifying template, 603–622
  - downloading a process template to your desktop, 601–602
  - navigating a process template, 602–603
  - overview, 601
- customization tools
  - Microsoft InfoPath 2003, 593–594
  - MSF HTML Build utility, 594
  - overview, 592–593
  - third-party tools, 594–595
  - XML/XSL editors, 593
- deleting, 623
- MSF for Agile Software Development, 482
- MSF for CMMI Process Improvement, 482
- New Team Project Wizard, 592
- overview, 591–592, 595, 603–605
- process guidance customization
  - activities, 597–598
  - cross-references, 600
  - custom roles, creating, 600
  - customizing work item elements, 599
  - overview, 595
  - ProcessGuidance.xml, 595–596
  - viewing your custom guidance, 601
  - work item documentation, 599
  - workstreams, 596–597
  - workstreams and activities, linking work items to, 599
- template definition modifications
  - classification, 619–621
  - core template, 605–607
  - groups, 621–622
  - overview, 603–605
  - permissions, 621–622
  - SQL Reporting Services, 617–619
  - Team Foundation version control, 607–609

- testing your changes, 613
- Windows SharePoint Services team portal, 613–617
- work items, 609–612
- testing your custom process templates, 623
- uploading process templates in Team Foundation Server, 622

ProcessGuidance.xml, **595–596**

### **Product Studio, 516**

**Professional UML with Visual Studio.NET:  
Unmasking Visio for Enterprise Architects  
(Wrox Press), 26**

### **profiling**

- ASP.NET profiling, 336
- code coverage, 335
- command-line tools
  - environment variables, setting, 331–332
  - executing a profiling session, 332
  - instrumentation configuration, 330–331
  - overview, 330
  - profiling a Windows service or ASP.NET application, 331
  - reports, generating, 332–333
  - VSIInstr.exe utility, 330–331
  - VSPerfClrEnv.cmd utility, 331–332
  - VSPerfCmd.exe utility, 332
  - VSPerfReport.exe utility, 332–333
- debugging symbols, 335
- described, 307
- virtual machines, 336

### **programming Application Verifier, 278–281**

### **prohibited bindings, 87**

### **project alerts, 511–513**

### **project, creation of, 133–134**

### **project files, 235–236**

### **project management**

- administering project details
  - deleting a Team project, 514–516
  - overview, 505
  - project alerts, 511–513
  - project security management, 507–508
  - project settings, configuring, 507–511
  - structuring your project, 508–510
  - Team Foundation Server, connecting to, 505–506

- Team Portal customization and extensibility, 513–514
- Team Project Portal, 511
- version control management, 510–511
- creating a software project
  - communication problems, 501
  - failure of software development projects, reasons for, 501–502
  - metrics and estimation skills, lack of, 501
  - New Team Project Wizard, 502–505
  - operational details, too much focus on, 501
  - overview, 500–501
  - poor workflow management, 501
  - standards compliance, lack of, 501
  - Team Project, 500
  - understanding of processes, lack of, 501
  - Visual Studio Project, 500
- overview, 499–500
- process guidance, 499
- reporting
  - custom reports, building, 530–532
  - Microsoft Excel pivot tables, 529–530
  - overview, 524
  - security for report site, 533
  - Team Reports, 524–529
- tools for managers, 499
- work items
  - access to work items, controlling, 518
  - anatomy of a work item, 516–517
  - Microsoft Excel used to create and assign work items, 519–521
  - Microsoft Office used to create and assign work items, 519–522
  - Microsoft Project used to create and assign work items, 521–522
  - overview, 516
  - queries, creating and running work item, 518–519
  - types, creating and customizing work item, 522–524
  - Visual Studio used to create work items, 517–518
- project management guidance, 483**
- project managers, 571, 587, 649**
- project security management, 507–508**

- project settings configuration**
  - overview, 507
  - project security management, 507–508
  - structuring your project, 508–510
  - Team Project Portal, 511
  - version control management, 510–511
- Promote Local Variable to Parameter, 296–297**
- properties of ordered tests, 353**
- Protection Location setting (Application Verifier), 265**
- ProtoGen, 216–217**
- prototypes**
  - described, 29
  - System Definition Model (SDM), 189–190
- provider endpoints, 31**
- Proxy attribute, 178**
- proxy support, 486**
- PSMs (platform-specific models), 5**
- publishing results, 351–352**

## Q

- QualifiedName attribute, 177**
- quality of service (QoS) requirement work item, 580–581**
- queries, 518–519**

## R

- RaiseError attribute, 183**
- Rational Rose, 22**
- reading session reports, 321–330**
- recording web tests, 397–398**
- recovery, 658–659**
- The recursion count is invalid error, 273**
- Redirect standard output and error to test result option (generic testing), 460**
- refactoring**
  - from Class Diagrams, 284–288
  - in code
    - Encapsulate Field, 291–294
    - Extract Method, 294–296
    - Generate Method Stub, 299–300
    - Promote Local Variable to Parameter, 296–297

### **refactoring (continued)**

- Remove Parameters, 298–299
- Rename, 288–291
- Reorder Parameters, 297–298
- Extract Interface, 284–287
- Implement Abstract Class, 287–288
- overview, 283–284
- Reference **attribute**, **178**
- ReferenceDefinition **relationship**, **176**
- referencing work products**, **597**
- reflection**, **237**
- The region of memory about to be freed contains an active lock error**, **273**
- registry**, **217–218**
- regression testing**
  - automated testing, 435
  - overview, 358
- REI (Rose Extensibility Interface)**, **22**
- relationship definition**, **170**
- RelationshipConstraints, **208–209**
- RelationshipDefinition **complex type**, **175–176**
- RelationshipMember **complex type**, **178**
- relationships**, **137–138**
- release managers**, **571, 589**
- Released memory containing an active lock error**, **272**
- Released virtual memory containing an active lock error**, **272**
- reliability rules**, **226**
- Remove Parameters**, **298–299**
- Rename**, **288–291**
- Reorder Parameters**, **297–298**
- Report Builder used to create custom reports**, **531–532**
- Report Definition Language files**, **530**
- reporting**
  - custom reports
    - building, 530–532
    - overview, 530–531
    - Report Builder used to create, 531–532
    - Report Definition Language files, 530
    - running, 532
    - viewing, 532

- deployment reports
  - automated deployment, 95–97
  - human-readable deployment reports, 94–95
  - machine-readable deployment reports, 95–97
  - overview, 93–94
- integration, 657
- Microsoft Excel pivot tables, 529–530
- migration, 657
- overview, 524
- security for report site, 533
- Team Foundation Build, 641–643
- Team Reports
  - overview, 524
  - Reporting Services Report Manager used for managing reports, 525–526
  - Team Explorer used for managing reports, 524
  - trends within a project, analyzing, 526
- Reporting Services Report Manager used for managing reports**, **525–526**
- reports**, **582**
- Required **attribute**, **179**
- results**
  - load tests
    - Errors table, 425–426
    - Graphs View of, 423–424
    - overview, 422–423
    - Pages table, 424
    - SQL Trace table, 425
    - Tables View of, 424–426
    - Tests table, 424
    - Thresholds table, 425
    - Transactions table, 424
  - overview, 422–423
  - unit testing, 367–368
  - viewing, 390–391
  - web tests, 402–403
- ReturnEarly **attribute**, **183**
- reusable prototypes**
  - overview, 77
  - sharing, 78
  - StockBrokerZone prototype, 77–78
- reverse engineering**
  - Application Designer, 53
  - described, 40

**revision field (work items), 517**  
**risk work item, 581–582**  
**Robbins, John (*Debugging Applications for Microsoft .NET and Microsoft Windows*), 266**  
**roles**  
 architect, 587–588  
 business analyst, 586–587  
 developer, 588  
 overview, 585–586  
 project manager, 587  
 release manager, 589  
 tester, 588–589  
 user experience, 589  
**Rose Extensibility Interface (REI), 22**  
**rule violations**  
 correcting problems, 230–232  
 overview, 230  
 suppressing messages, 232–233  
**rules for Managed Code Analysis, built-in, 225–226**  
 Rules.xml, **242–243**  
**run settings (Load Test Editor), 420**  
**Run Settings (New Load Test Wizard), 417–418**

## S

**SA\_FormatString attribute, 254**  
**SA\_InvalidCheck attribute, 254**  
**sample application creation, 309–310**  
**sample logical datacenter**  
 datacenter zones, 63  
 DMZ1 zone, 63  
 DMZ2 zone, 63  
 endpoint styles, 64  
 IntranetZone, 63  
 overview, 62–63  
 UML deployment diagram, 64–65  
**sample web application creation**  
 configuring sample application for testing, 396  
 overview, 394  
 users for site, creating, 395–396  
**sampling profilers, 308**  
**sampling session configuration, 318–319**  
 SA\_No attribute, 254  
 SA\_NoAccess attribute, 254  
 SA\_Post attribute, 254  
 SA\_Pre attribute, 254  
 SA\_Read attribute, 254  
 SA\_ReadWrite attribute, 254  
 SA\_Write attribute, 254  
 SA\_Yes attribute, 254  
**scenario work item, 581**  
**schema structure, 169–170**  
**SCVs (special cause variations), 569**  
**SDLC (Software Development Lifecycle), 22**  
**SDM Compiler, 163, 191–192, 216**  
**SDM documents, 162, 186–189**  
**SDM files, 42–43**  
**SDM SDK, 190–191**  
**SDM (System Definition Model)**  
 abstract attribute, 172  
 abstract objects, 161  
 abstract relationships, 161  
 advantages of, 160  
 architecture for, 160–163  
 complex types, 167–168  
 constraints  
   ConstraintDefinition complex type, 182–183  
   ConstraintMember complex type, 183–184  
   Evaluate attribute, 183  
   FlowMember complex type, 184–185  
   ManagerDeclaration complex type, 185–186  
   overview, 182  
   RaiseError attribute, 183  
   ReturnEarly attribute, 183  
   setting constraints, 182  
   structural constraints, 182  
   TargetDefinition attribute, 183  
 Definition complex type, 170–171  
 definition members, 170  
 deployment, 165–166  
 design-time validation, 163–164  
 Distributed System Designers, 189–190

### **SDM (System Definition Model) (continued)**

#### example

- Application layer endpoint, 196–198
- ApplicationHost layer endpoint, 199–200
- communication, 200–201
- constraints, 208–216
- containment, 201–205
- delegation, 200–201
- endpoints, 196–200
- flows, 207
- hosting, 206
- overview, 192–193
- ProtoGen, 216–217
- registry, 217–218
- SDM Compiler, 216
- settings, 195–196
- settings for endpoints, 198–199
- SystemDefinition, 193–195
- Visual Studio and, 218–219

extends attribute, 172

instances, 161–162

layer attribute, 172

managers, 162

Member complex type, 176–179

modeling layers of a system, 166–167

object definition, 170

ObjectDefinition complex type, 172–175

overview, 22, 160

prototypes, 189–190

relationship definition, 170

RelationshipDefinition complex type, 175–176

schema structure, 169–170

SDM Compiler, 163, 191–192

SDM documents, 162, 186–189

SDM SDK, 190–191

settings

- Access attribute, 179
- custom settings, 181–182
- DeploymentTime attribute, 179
- overview, 179–180
- Required attribute, 179
- Secure attribute, 179
- setting values, 180–181

- simple types, 167–168
- system deployment definition, 166
- Team System meta-model, 22–23
- type definitions, 168
- types, 167–169
- in Visual Studio 2005 Team Edition for Software Architects, 164–165

### **SdmG (SDM Manager Generator), 212–215**

Secure attribute, 179

#### **security**

- groups, 621–622
- for report site, 533
- roles, 542–543
- rules, 226
- setting up build, 630

### **selecting tests, 343–344**

### **Semantic Model Definition Language (SMDL) model, 531**

### **Server Administrator, 649**

#### **session reports**

- Allocation View, 328
- application time, 322
- Call Tree View, 327
- Caller/Callee View, 326–327
- elapsed time, 322
- exclusive time, 322
- Functions View, 325–326
- inclusive time, 322
- interpreting, 321–330
- managing, 321
- Objects Lifetime View, 328–330
- reading, 321–330
- Summary View, 322–325

### **session targets, configuring, 316–318**

### **setting check-in notes for your C/C++ code, 259–260**

### **setting constraints, 182**

### **setting values, 180–181**

### **setting warning levels, 249–250**

SettingMembers complex type, 178

#### **settings**

- Access attribute, 179
- custom settings, 181–182
- DeploymentTime attribute, 179
- example (SDM), 195–196

- overview, 179–180
- Required attribute, 179
- Secure attribute, 179
- setting values, 180–181
- System Definition Model (SDM), 179–182
- web tests, 400–401
- settings and constraints**
  - constraints versus settings, 71–72
  - custom settings, 75
  - GenericServer settings and constraints, 72
  - IISWebServer settings and constraints, 72–74
  - Import IIS Settings Wizard, 76
  - overview, 71
  - user-defined constraints, 76
  - WebSiteEndpoint settings and constraints, 74
  - ZoneEndpoint settings and constraints, 74–75
- Settings and Constraints Editor, 53–54**
- settings for endpoints, 198–199**
- setup**
  - Application Verifier, 264–266
  - Team Foundation Build
    - build lab, setting up, 629
    - build location folder, 629
    - drop location folder, 629
    - overview, 627–628
    - security, setting up build, 630
    - strategy for build, planning a, 628
    - Team System integration, 629–630
  - Team Foundation version control
    - overview, 541–542
    - security roles, 542–543
    - workspace, 543–544
- severe bugs, 438**
- Shape Map, providing a, 144–145**
- shapes, defining, 142–143**
- sharing reusable prototypes, 78**
- shelvesets, 557–559**
- shelving, 486**
- Short, Keith (*Software Factories: Assembling Applications with Patterns, Works, Models and Tools*), 5, 7**
- Simple Object Access Protocol (SOAP), 11**
- simple types, 167–168**
- `SimpleName` attribute, 177
- simulating collaboration diagrams using comments, 127**
- single-server deployment, 652**
- sites, 135–137**
- small to medium deployments, 650**
- SMDL (Semantic Model Definition Language) model, 531**
- Software Development Lifecycle (SDLC), 22**
- software factories, 7–8**
- Software Factories: Assembling Applications with Patterns, Works, Models and Tools (Short and Greenfield)*, 5, 7**
- Software Factories Workbench, 8**
- sorting test results, 350**
- sorting tests, 346**
- source code migration, 540–541**
- source control and item tracking, 21**
- Source Control Explorer**
  - overview, 544–545
  - source repository, adding projects to, 548
  - workspaces, 546–548
- source repository, adding projects to, 548**
- special cause variations (SCVs), 569**
- special-purpose test lists, 345**
- Specify an existing program to wrap as a generic test option, 459**
- SQL Reporting Services, 617–619**
- SQL Trace table, 425**
- SQL Tracing, 420**
- standards compliance, lack of, 501**
- Standish Group (CHAOS report), 481**
- starting a build, 638**
- `StartsWith` method, 375
- state field (work items), 517**
- state pattern used to simulate statechart diagrams, 125–127**
- static code analysis, 246. See also Code Analysis for C/C++**
- status, 637–638**
- status field (work items), 516**

### Stockbroker example

- Application Designer
  - database connection properties, specifying, 36–37
  - design characteristics, 33–34
  - drawing the diagram, 34–36
  - endpoint details, specifying, 37–38
  - endpoint prototype, defining your own, 38–39
  - existing service, representing an, 39–40
  - overview, 32–33
  - UML comparisons, 34
- Class Designer, 104–108
- DealingService.vb, 46–47
- Logical Datacenter Designer
  - Client zone, 70
  - MarketMaker zone, 65–67
  - overview, 65
  - StockBroker zone, 68–69
  - StockMarket zone, 67–68
- overview, 18, 45
- StockQuoteService.vb, 46
- testing StockBroker application, 47–48
- StockBroker zone, 68–69**
- StockBrokerZone prototype, 77–78**
- StockDeal classes, 105–108, 110–115**
- StockMarket zone, 67–68**
- StockQuote structure, 117**
- StockQuoteApp implementation**
  - Default.aspx file, 49
  - Default.aspx.cs file, 49–50
  - overview, 48
  - testing StockQuoteApp Web application, 50
- StockQuoteService.vb, 46**
- stopping a build, 639**
- storing load test run data, 422**
- strategy for build, planning a, 628**
- stress testing, 435**
- StringAssert **class**
  - Contains method, 375
  - DoesNotMatch method, 375
  - EndsWith method, 375
  - Matches method, 375
  - overview, 375–376
  - StartsWith method, 375

- structural constraints, 182**
- structural testing, 435**
- structure, 448–449**
- Structure type, 103**
- structuring your project, 508–510**
- summary report XML schema, 468–469**
- Summary View for session reports, 322–325**
- suppressing messages, 232–233**
- system definition for a subset of the application design, 89–90**
- system definition for the full application design, 89**
- System Definition Model (SDM)**
  - abstract attribute, 172
  - abstract objects, 161
  - abstract relationships, 161
  - advantages of, 160
  - architecture for, 160–163
  - complex types, 167–168
  - constraints
    - ConstraintDefinition complex type, 182–183
    - ConstraintMember complex type, 183–184
    - Evaluate attribute, 183
    - FlowMember complex type, 184–185
    - ManagerDeclaration complex type, 185–186
    - overview, 182
    - RaiseError attribute, 183
    - ReturnEarly attribute, 183
    - setting constraints, 182
    - structural constraints, 182
    - TargetDefinition attribute, 183
  - Definition complex type, 170–171
  - definition members, 170
  - deployment, 165–166
  - design-time validation, 163–164
  - Distributed System Designers, 189–190
  - example
    - Application layer endpoint, 196–198
    - ApplicationHost layer endpoint, 199–200
    - communication, 200–201
    - constraints, 208–216
    - containment, 201–205

- delegation, 200–201
- endpoints, 196–200
- flows, 207
- hosting, 206
- overview, 192–193
- ProtoGen, 216–217
- registry, 217–218
- SDM Compiler, 216
- settings, 195–196
- settings for endpoints, 198–199
- SystemDefinition, 193–195
- Visual Studio and, 218–219
- extends attribute, 172
- instances, 161–162
- layer attribute, 172
- managers, 162
- Member complex type, 176–179
- modeling layers of a system, 166–167
- object definition, 170
- ObjectDefinition complex type, 172–175
- overview, 22, 160
- prototypes, 189–190
- relationship definition, 170
- RelationshipDefinition complex type, 175–176
- schema structure, 169–170
- SDM Compiler, 163, 191–192
- SDM documents, 162, 186–189
- SDM SDK, 190–191
- settings
  - Access attribute, 179
  - custom settings, 181–182
  - DeploymentTime attribute, 179
  - overview, 179–180
  - Required attribute, 179
  - Secure attribute, 179
  - setting values, 180–181
- simple types, 167–168
- system deployment definition, 166
- and the Team System meta-model, 22–23
- type definitions, 168
- types, 167–169
- in Visual Studio 2005 Team Edition for Software Architects, 164–165

- System Definition Model (SDM) files, 42–43**
- system deployment definition, 166**
- system deployments, 88–89**

### System Designer

- Comment tool, 82
- Connection tool, 82
- creating system diagrams from scratch, 92
- defining and deploying systems
  - defining deployment, 90–91
  - nested systems, 91–92
- overview, 89
- system definition for a subset of the application design, 89–90
- system definition for the full application design, 89
- dynamic modeling, 98
- overview, 15–16, 82
- system settings and constraints, 92–93
- System View, 83
- toolbox, 82

### System View

- Application Designer, 85
- Deployment Designer, 83
- System Designer, 83

## T

- Tables View of results, 424–426**

- TargetDefinition attribute, 183

- task work item, 582**

- TDD (test-driven development), 359–361**

### Team Developer profiler

- instrumentation session configuration, 319–320
- overview, 309
- Performance Explorer
  - Counters property page, 315
  - Events property page, 315
  - General property page, 314–315
  - general session properties, setting, 314–316
  - Launch property page, 315
  - overview, 313–314
  - session targets, configuring, 316–318
- performance session creation

### **Team Developer profiler (continued)**

- blank performance session, adding, 312
- overview, 311
- unit test, creating a performance session from a, 312–313
- using Performance Wizard, 311–312
- performance session execution, 320
- sample application creation, 309–310
- sampling session configuration, 318–319
- session reports
  - Allocation View, 328
  - application time, 322
  - Call Tree View, 327
  - Caller/Callee View, 326–327
  - elapsed time, 322
  - exclusive time, 322
  - Functions View, 325–326
  - inclusive time, 322
  - interpreting, 321–330
  - managing, 321
  - Objects Lifetime View, 328–330
  - reading, 321–330
  - Summary View, 322–325

### **Team Edition Test Load Agent, 489**

#### **Team Explorer**

- overview, 490–494
- used for managing reports, 524

#### **Team Foundation architecture**

- clients, 490
- configuration, 496–497
- overview, 489–490
- Team Explorer, 490–494
- Team Foundation Build, 494
- Team Foundation Reporting Warehouse architecture, 494–496

#### **Team Foundation Build**

- build types creation
  - deleting a build type, 635
  - editing an existing Team build type, 633–635
- New Team Build Type Creation Wizard used for, 631–633
- overview, 630–631

- command-line interface
    - deleting a build, 638–639
    - nightly builds, setting up, 639–640
    - overview, 638
    - starting a build, 638
    - stopping a build, 639
  - Continuous Integration (CI), 640–641
  - customization, 644–646
  - e-mail notifications, 635–636
  - extensibility, 644–646
  - local build, executing, 636–637
  - MSBuild tasks, 645–646
  - overview, 625–627, 630–631
  - reporting, 641–643
  - setup
    - build lab, setting up, 629
    - build location folder, 629
    - drop location folder, 629
    - overview, 627–628
    - security, setting up build, 630
    - strategy for build, planning a, 628
    - Team System integration, 629–630
  - status, 637–638
  - Team Foundation architecture, 494
  - Team Foundation Server, 488
- ### **Team Foundation Reporting, 488–489**
- #### **Team Foundation Reporting Warehouse architecture, 494–496**
- #### **Team Foundation Server**
- connecting to, 505–506
  - core services, 489
  - features, 483–484
  - integration
    - check-in policies using C/C++ code analysis, creating, 257–259
    - overview, 257
    - setting check-in notes for your C/C++ code, 259–260
    - work items created from code analysis warnings, 260–261
  - Microsoft Solutions Framework (MSF), 482
  - overview, 482
  - project management guidance, 483
  - publishing test results on, 452–453
  - Team Edition Test Load Agent, 489

- Team Foundation Build, 488
- Team Foundation Reporting, 488–489
- team processes, 482
- Team Project Portal, 488
- version control
  - branching, 485
  - check-in policies, 486–487
  - history, 485
  - integrated check-in, 484
  - merging, 485
  - migrating from VSS to Team Foundation version control, 487
  - overview, 484
  - proxy support, 486
  - shelving, 486
  - workspaces, 485
- work item tracking (WIT), 487–488
- Team Foundation version control**
  - administrator role, 543
  - branching, 553–555
  - check-in and check-outs
    - checking in an item, 549
    - checking out an item, 550–551
  - policies, creating and administering
    - check-in, 551–553
  - command-line tools, 560–561
  - contributor role, 543
  - development role, 543
  - features of, 537
  - merging, 555–557
  - overview, 535–536
  - process templates, 607–609
  - setup for
    - overview, 541–542
    - security roles, 542–543
    - workspace, 543–544
  - shelvesets, 557–559
  - source code migration, 540–541
  - Source Control Explorer
    - overview, 544–545
    - source repository, adding projects to, 548
    - workspaces, 546–548
  - test lead role, 543
  - Visual SourceSafe (VSS) 2005 compared, 536–539

**Team Model**

- advocacy, 584–585
- mindsets, 585
- overview, 582–584

**Team Portal customization and extensibility, 513–514****team processes, 482****Team Project, 500****Team Project Portal, 488, 511****Team Reports**

- overview, 524
- Reporting Services Report Manager used for managing reports, 525–526
- Team Explorer used for managing reports, 524
- trends within a project, analyzing, 526529

**Team System**

- integration, 629–630
- manual testing
  - bug work item, logging your manual test as a, 451–452
  - configuring manual test options, 440–442
  - creating manual tests, 439
  - managing manual tests, 449–453
  - overview, 438–439
  - structure, 448–449
  - Team Foundation Server, publishing test results on, 452–453
  - templates, 442–448
  - Test Manager, 450–451
- manual testing templates
  - adding a custom manual test in your project, 443
  - creating a Visual Studio 2005 test template, 446
  - creating custom manual test templates, 443–445
  - default templates, 442–443
  - exporting your custom manual test as a new item, 443–445
  - integrating your custom manual test template into Visual Studio 2005, 446–447
  - Microsoft Word format template, 447–448

### **Team System (continued)**

- opening test templates using Open With Wizard, 447
- overview, 442–443
- Text Format template, 448
- obtaining, 659–660
- profiling, 308

### **Team System meta-model and System Definition Model (SDM), 22–23**

#### **Team System unit testing**

- creating your first unit test
- failing tests, 363
- identifying unit test classes, 363
- identifying unit tests, 363
- overview, 362–363
- successful tests, 363
- debugging unit tests, 368
- managing and running unit tests
- overview, 364
- Test Manager, 365–366
- Test View, 365
- overview, 361–362
- results, 367–368
- test run configuration, 366–367

#### **template definition modifications**

- process templates
- classification, 619–621
- core template, 605–607
- groups, 621–622
- overview, 603–605
- permissions, 621–622
- SQL Reporting Services, 617–619
- Team Foundation version control, 607–609
- testing your changes, 613
- Windows SharePoint Services team portal, 613–617
- work items, 609–612
- Windows SharePoint Services team portal
- adding new work products to, 615
- custom logo added to, 615
- customizing, 615
- document template schema, 613–614
- integrating a site template within a process template, 617
- overview, 613

- saving your custom site template, 616–617
- verifying your portal customizations, 617

#### **templates**

- adding a custom manual test in your project, 443
- creating a Visual Studio 2005 test template, 446
- creating custom manual test templates, 443–445
- default templates, 442–443
- exporting your custom manual test as a new item, 443–445
- integrating your custom manual test template into Visual Studio 2005, 446–447
- Microsoft Word format template, 447–448
- opening test templates using Open With Wizard, 447
- overview, 442–443
- Text Format template, 448

#### **test case management**

- ordered tests
- analyzing, 354–355
- creating, 352–353
- executing, 354
- overview, 352
- properties of, 353
- overview, 342
- Test Manager
- columns displayed, choosing, 345–346
- disabling tests, 344
- filtering tests, 346–347
- grouping tests, 346
- overview, 342–343
- running tests, 344
- selecting tests, 343–344
- sorting tests, 346
- test lists, 344–345
- test meta-data files, loading and importing, 347–348
- Test Results window
- exporting test results, 350
- filtering test results, 350
- grouping test results, 350
- overview, 349–350
- publishing results, 351–352

- sorting test results, 350
  - Test Results XML (TRX) files, 350–351
  - Test View window, 348–349
- test cases, 437**
- test deployment using virtualization, 654**
- test lead role, 543**
- test lists**
  - creating, 344–345
  - overview, 344
  - populating, 345
  - special-purpose test lists, 345
  - Test Manager, 344–345
  - working with, 345
- Test Manager**
  - columns displayed, choosing, 345–346
  - disabling tests, 344
  - filtering tests, 346–347
  - grouping tests, 346
  - overview, 342–343
  - running tests, 344
  - selecting tests, 343–344
  - sorting tests, 346
  - test lists
    - creating, 344–345
    - overview, 344
    - populating, 345
    - special-purpose test lists, 345
    - working with, 345
  - test meta-data files, loading and importing, 347–348
- Test Mix, 413–415**
- test projects**
  - creating, 340–341
  - generic tests, 340
  - load tests, 340
  - manual tests, 340
  - metadata file, 341
  - options, setting, 341–342
  - ordered tests, 340
  - overview, 339–340
  - test run configuration, 341
  - types of tests in, 340
  - unit tests, 340
  - web tests, 340
- Test Results window**
  - exporting test results, 350
  - filtering test results, 350
  - grouping test results, 350
  - overview, 349–350
  - publishing results, 351–352
  - sorting test results, 350
  - Test Results XML (TRX) files, 350–351
- test rig considerations, 655**
- test run configuration**
  - distributed load tests, 429–430
  - Team System unit testing, 366–367
  - test projects, 341
- Test View window, 348–349**
- `TestContext` **class, 377–378**
- test-driven development (TDD), 359–361**
- testers, 571, 588–589**
- testing**
  - DealingApp Windows application, 51–52
  - MarketMaker application, 44–45
  - StockBroker application, 47–48
  - StockQuoteApp Web application, 50
  - your changes to process templates, 613
  - your custom process templates, 623
- testing methodologies (Hunter), 453**
- testing tools**
  - integration, 657–658
  - migration, 657–658
- `TestProperty` **attribute, 377**
- Tests table, 424**
- Text Format template, 448**
- think times, 412**
- third-party tools, 361, 594–595**
- The thread cannot own a lock in this context error, 272**
- Thread not supposed to be owning a lock error, 273**
- Thresholds table, 425**
- title field (work items), 516**
- toolbox**
  - Application Designer
    - applications, 31–32
    - ASPNETWebApplication, 32
    - ASPNETWebService, 32

### **toolbox (continued)**

- BizTalkWebService, 32
- comment tool, 31
- connection tool, 31
- endpoints, 31
- ExternalDatabase, 32
- ExternalWebService, 32
- General Designer, 31
- GenericApplication, 32
- GenericEndpoint, 31
- OfficeApplication, 32
- overview, 30
- pointer tool, 31
- WebContentEndpoint, 31
- WebServiceEndpoint, 31
- WindowsApplication, 32

Class Designer

- Abstract Class type, 103
- Association item, 103
- Class type, 102
- Comment tool, 103
- Delegate type, 103
- Enum type, 103
- Inheritance item, 103
- Interface type, 103
- Module type, 103
- overview, 102
- Structure type, 103

defining, 144

Logical Datacenter Designer

- Comment tool, 60
- Connection tool, 60
- DatabaseClientEndpoint, 61
- DatabaseServer, 61
- endpoints, 61
- General Designer, 60
- GenericClientEndpoint, 61
- GenericServer, 61
- GenericServerEndpoint, 61
- HTTPClientEndpoint, 61
- IISWebServer, 61
- Logical Servers, 61
- overview, 60
- WebSiteEndpoint, 61

- WindowsClient, 61
- Zone, 61
- ZoneEndpoint, 61
- System Designer, 82

**tools for managers, 499**

**tracing markers, 308**

**tracks, 577–578**

**Transactions table, 424**

**translation code, 155**

**trends within a project, analyzing, 526–529**

**TRX (Test Results XML) files, 350–351**

**type definitions, 168**

**type field (work items), 516**

**types**

- complex types, 167–168
- creating and customizing work item, 522–524
- described, 101–102
- other than class, 116–118
- of profilers, 308
- simple types, 167–168
- of tests in test projects, 340

## **U**

**UCL (Upper Control Limits), 569**

**UML (Unified Modeling Language)**

- Application Designer, UML comparisons, 34
- Class Designer, UML class diagrams compared, 128–129
- component diagram, 10
- Deployment Designer, UML deployment diagram compared, 97–98
- deployment diagram, 64–65
- existing models and, 24
- fully integrated UML capability, achieving, 24–25
- new visual designers, comparisons to, 23–24
- overview, 23

**UNC (Universal Naming Convention), 629**

**understanding of processes, lack of, 501**

**Unexpected exception raised in heap code path error, 269**

**unit test, creating a performance session from, 312–313**

**Unit Test Framework**

## Assert methods

- Assert.AreEqual method, 371–372
- Assert.AreNotEqual method, 371–372
- Assert.AreNotSame method, 372–373
- Assert.AreSame method, 372–373
- Assert.Fail method, 374
- Assert.Inconclusive method, 374
- Assert.IsFalse method, 373
- Assert.IsInstanceOfType method, 373
- Assert.IsNotInstanceOfType method, 373
- Assert.IsNotNull method, 373
- Assert.IsNull method, 373
- Assert.IsTrue method, 373

## initialization and cleanup of unit tests

- AssemblyCleanup attribute, 371
- AssemblyInitialize attribute, 371
- ClassCleanup attribute, 370–371
- ClassInitialize attribute, 370–371
- overview, 368–369
- TestCleanup attribute, 369–370
- TestInitialize attribute, 369–370

## unit testing

- AllItemsAreInstancesOfType method, 374
- AllItemsAreNotNull method, 374
- AllItemsAreUnique method, 374
- AreEqual method, 374
- AreEquivalent method, 374
- AreNotEqual method, 374
- AreNotEquivalent method, 374
- Assert methods, 371–374
- CollectionAssert class, 374–375
- Contains method, 374, 375
- custom unit test properties, defining, 377
- data-driven unit tests, creating, 378
- DoesNotContain method, 374
- DoesNotMatch method, 375
- EndsWith method, 375
- exceptions, 376
- ExpectedException attribute, 376
- initialization and cleanup of unit tests, 368–371

- IsNotSubsetOf method, 374
- IsSubsetOf method, 374
- Matches method, 375
- StartsWith method, 375
- StringAssert class, 375–376
- TestContext class, 377–378
- TestProperty attribute, 377

**unit testing**

- advantages of, 358–359
- ASP.NET applications
  - attributes of ASP.NET unit tests, 387–388
  - creating ASP.NET unit tests, 388–389
  - overview, 387
- code coverage
  - enabling, 390
  - overview, 389–390
  - results, viewing, 390–391
- code generation
  - generating code from tests, 386–387
  - generating tests from code, 382–386
  - overview, 382
- with NUnit, 361
- overview, 358
- private members testing
  - overview, 379
  - PrivateObject class, 379–381
  - PrivateType class, 381–382
- Team System unit testing
  - creating your first unit test, 362–364
  - debugging unit tests, 368
  - managing and running unit tests, 364–366
  - overview, 361–362
  - results, 367–368
  - test run configuration, 366–367
- test projects, 340
- test-driven development (TDD), 359–361
- third-party tools for, 361
- Unit Test Framework
  - AllItemsAreInstancesOfType method, 374
  - AllItemsAreNotNull method, 374
  - AllItemsAreUnique method, 374
  - AreEqual method, 374
  - AreEquivalent method, 374

### **unit testing (continued)**

- AreNotEqual method, 374
  - AreNotEquivalent method, 374
  - Assert methods, 371–374
  - CollectionAssert class, 374–375
  - Contains method, 374, 375
  - custom unit test properties, defining, 377
  - data-driven unit tests, creating, 378
  - DoesNotContain method, 374
  - DoesNotMatch method, 375
  - EndsWith method, 375
  - exceptions, 376
  - ExpectedException attribute, 376
  - initialization and cleanup of unit tests, 368–371
  - IsNotSubsetOf method, 374
  - IsSubsetOf method, 374
  - Matches method, 375
  - StartsWith method, 375
  - StringAssert class, 375–376
  - TestContext class, 377–378
  - TestProperty attribute, 377
  - writing unit tests, 359
- Universal Naming Convention (UNC), 629**
- unsupported deployment scenarios, 650**
- uploading process templates in Team Foundation Server, 622**
- Upper Control Limits (UCL), 569**
- usage rules, 226**
- user accounts, 651**
- user experience, 589**
- user-defined constraints, 76**
- Users, 649**

## **V**

### **validating the deployment, 87–88**

#### **validation**

- constraints, 209–211
- web test rules, 405–407

#### **Verification Layer settings (Application Verifier), 266**

#### **version control**

- integration, 656
- management, 510–511

- migration, 656
- Team Foundation Server
  - branching, 485
  - check-in policies, 486–487
  - history, 485
  - integrated check-in, 484
  - merging, 485
  - migrating from VSS to Team Foundation version control, 487
  - overview, 484
  - proxy support, 486
  - shelving, 486
  - workspaces, 485

#### **versions**

- Application Verifier, 263
- Code Analysis for C/C++, 246

#### **viewing**

- code analysis warnings and errors, 250–251
- custom guidance, 601
- custom reports, 532
- distributed load tests, 430

#### **virtual machines, 336**

#### **Visio for Enterprise Architects, 26**

#### **visual designers**

- Application Designer, 13–14
- Class Designer, 16–17
- Deployment Designer, 15–16
- Logical Datacenter Designer, 14–15
- overview, 12–13
- System Designer, 15–16
- UML compared to, 23–24

#### **Visual SourceSafe (VSS) 2005, 536–539**

#### **Visual Studio 2005 integration**

- code highlighting, 250–251
- disabling C/C++ code analysis, 248–249
- enabling C/C++ code analysis, 248–249
- overview, 248
- setting warning levels, 249–250
- viewing code analysis warnings and errors, 250–251

#### **Visual Studio 2005 Team Edition for Software Architects, 164–165**

#### **Visual Studio Project, 500**

#### **Visual Studio used to create work items, 517–518**

**visualization of existing code**

- DealingApp class diagram, 104–105
- members, visualizing, 108–109
- overview, 103
- StockBroker example, 104–108
- StockDeal classes, 105–108

**VSS (Visual SourceSafe) 2005, 536–539****W****Web Services Description Language (WSDL)****file, 32****web tests**

- automated testing, 435
- coded web tests, 409–411
- configuring web test run settings, 399
- context parameters, 404–405
- creating, 396–397
- data-driven web tests
  - binding to a source, 408–409
  - data source configuration, 407–408
  - overview, 407
- editing
  - comments, 403–404
  - overview, 403
  - request properties, 403
  - transactions, 404
- execution of, observing, 402–403
- extraction rules, 404–405
- overview, 394
- parameterizing the web server, 399–400
- recording, 397–398
- results, observing, 402–403
- running, 401–402
- sample web application creation
  - configuring sample application for testing, 396
  - overview, 394
  - users for site, creating, 395–396
- settings, 400–401
- test projects, 340
- validation rules, 405–407

**WebContentEndpoint (Application Designer), 31****WebServiceEndpoint (Application Designer), 31****WebSiteEndpoint (Logical Datacenter Designer), 61****WebSiteEndpoint settings and constraints, 74****white box testing, 358****Windows applications and services, 56****Windows Scripting Host, 464–465****Windows SharePoint Services team portal**

- process templates, 613–617
- template definition modifications
  - adding new work products to, 615
  - custom logo added to, 615
  - customizing, 615
  - document template schema, 613–614
  - integrating a site template within a process template, 617
  - overview, 613
  - saving your custom site template, 616–617
  - verifying your portal customizations, 617

**WindowsApplication (Application Designer), 32****WindowsClient (Logical Datacenter Designer), 61****wiring to existing test systems**

- generic testing
  - JScript conversion code, 473–477
  - JUnit, creating a wire to, 471–473
  - overview, 470–471
- JUnit, creating a wire to
  - flow of operation, 471–472
  - overview, 471
  - setup and run, 472–473

**work item tracking (WIT)**

- access to work items, controlling, 518
- integration, 657
- migration, 657
- overview, 516
- queries, creating and running work item, 518–519
- Team Foundation Server, 487–488
- types, creating and customizing work item, 522–524

**work item types**

- bug work item, 579–580
- overview, 578–579
- quality of service (QOS) requirement work item, 580–581

## work item types (continued)

---

### **work item types (continued)**

risk work item, 581–582

scenario work item, 581

task work item, 582

### **work items**

anatomy of a work item, 516–517

assigned to field, 517

created from code analysis warnings,  
260–261

documentation, 599

ID field, 516

instances, 612

Microsoft Excel used to create and assign  
work items, 519–521

Microsoft Office used to create and assign  
work items, 519–522

Microsoft Project used to create and assign  
work items, 521–522

process templates, 609–612

queries, 574–575, 611–612

revision field, 517

state field, 517

status field, 516

title field, 516

type field, 516

Visual Studio used to create work items,  
517–518

### **work products, 574**

### **workspaces, 485, 543–544, 546–548**

### **workstreams**

and activities, linking work items to, 599

associating activities to, 598

Microsoft Solutions Framework (MSF),  
589–590

overview, 596

process guidance customization

assigning roles to, 596

creating new workstreams, 596

entry and exit criteria in, 596–597

overview, 596

process templates, 596–597

### **writing the implementation code, 240–241**

### **writing unit tests, 359**

**Wrox Press (*Professional UML with Visual Studio.NET: Unmasking Visio for Enterprise Architects*), 26**

**WSDL (Web Services Description Language) file, 32**

## **X**

**XML Metadata Interchange (XMI), 22**

**XML/XSL editors, 593**

## **Z**

**Zone (Logical Datacenter Designer), 61**

**ZoneEndpoint**

overview, 61

settings and constraints, 74–75











