

Contents

About the Authors	vii
Acknowledgments	ix
Introduction	xxv
Part One: Team Architect	1
Chapter 1: Introducing the Visual Designers	3
Why Design Visually?	4
Microsoft's Modeling Strategy	5
Model-driven development	5
Domain-specific languages	6
Software Factories	7
From Objects to Services	8
Objects and compile-time reuse	8
Components and deploy-time reuse	9
Distributed components and run-time reuse	10
Distributed services and the service-oriented architecture	11
Visual Designers	12
Application Designer	13
Logical Datacenter Designer	14
System Designer and Deployment Designer	15
Class Designer	16
Code synchronization	17
Introducing the StockBroker Case Study	18
Designer Relationships and Team System Integration	19
Process flow (roles and activities)	19
Object flow (artifacts)	20
Settings and constraints	21
Source control and item tracking	21
Introducing the System Definition Model	22
Meta-models	22
SDM and the Team System meta-model	22
What about UML?	23
Capitalizing on your investment in UML skills	23
Capitalizing on your investment in UML artifacts	24

Contents

Achieving a fully integrated UML capability	24
A retained role for Visio for Enterprise Architects	26
Summary	27
Chapter 2: Application Modeling Using Application Designer	29
Application Designer Toolbox and Prototypes	30
General Designer	31
Endpoints	31
Applications	31
Designing the Distributed System	32
The Design scenario	32
Design characteristics and UML comparison	33
Reproducing the design using Application Designer	34
Implementing the Applications	41
A note about SDM files	42
The MarketMaker implementation	43
The StockBroker implementation	45
The StockQuoteApp implementation	48
The DealingApp implementation	50
Application Designer Additional Features	52
Model and code synchronization	52
Reverse engineering existing solutions	53
Settings and Constraints Editor	53
Printing and copying diagrams	54
Application Designer Limitations and Workarounds	54
Modeling .NET Remoting and DCOM applications	55
Modeling Windows services and console applications	56
Dynamic Modeling	56
Summary	57
Chapter 3: Logical Infrastructure Modeling Using Logical Datacenter Designer	59
Logical Datacenter Designer Toolbox and Prototypes	60
General	60
Endpoints	61
Logical servers	61
Sample Logical Datacenter	62
Datacenter zones	63
Endpoint styles	64
The UML Deployment Diagram: similar, but not the same	64

Defining a Logical Datacenter for the StockBroker Applications	65
The MarketMaker Zone	65
The StockMarket Zone	67
The StockBroker Zone	68
The Client Zone	70
Settings and Constraints	71
Constraints versus settings	71
GenericServer settings and constraints	72
IISWebServer settings and constraints	72
WebSiteEndpoint settings and constraints	74
ZoneEndpoint settings and constraints	74
Custom settings and user-defined constraints	75
Import IIS Settings Wizard	76
Defining Your Own Reusable Prototypes	77
StockBrokerZone prototype	77
Team-working with prototypes	78
Summary	78
Chapter 4: Defining Systems and Evaluating Deployments Using System Designer and Deployment Designer	81
Introducing System Designer and Deployment Designer	82
Introducing System Designer	82
Introducing Deployment Designer	82
Toolbox and System View	82
Defining a Default Deployment from Application Designer	83
The System View	85
Binding applications to servers	85
Differences between default deployments and system deployments	88
Defining and Deploying Systems	89
System definition for the full application design	89
System definition for a subset of the application design	89
Defining deployment	90
Nested systems	91
Creating System Diagrams from Scratch	92
System Settings and Constraints	92
Deployment Reports	93
Human-readable deployment reports	94
Machine-readable deployment reports and automated deployment	95
The UML Deployment Diagram: Similar, but Not the Same	97
Dynamic Modeling	98
Summary	98

Chapter 5: Class Modeling Using Class Designer	101
Class Designer Toolbox and Types	102
From Code to Class Diagrams	103
StockBroker revisited — from code to class diagrams	104
Visualizing members	108
From Class Diagrams to Code	109
Designing the StockDeal classes using Class Designer	110
Working with methods	115
Working with other types	116
Working with interfaces	118
Reproducing class diagrams in hardcopy	119
Advanced Topics	120
Code synchronization	120
.NET Framework classes on Class Diagrams	121
Patterns	121
Building your own pattern library	122
Dynamic modeling	125
Comparison with UML Class Diagrams	128
Summary	129
Chapter 6: DSL Tools	131
Domain-Specific Languages	131
Creating a Designer	132
Creating the Project	133
Creating the Navigation Language	134
Creating the Graphical Language	141
Defining the shapes	142
Defining the connectors	143
Defining the toolbox	144
Providing a Shape Map	144
Providing a Connector Map	145
Using the Designer	147
Using the Domain Model	149
Generating Code	151
Designers and VSTS	154
Summary	156
Chapter 7: Dynamic Systems Initiative and the System Definition Model	157
Dynamic Systems Initiative	159
System Definition Model	160
SDM Architecture	160
Design-time validation	163

SDM in Visual Studio 2005 Team Edition for Software Architects	164
Deployment	165
Modeling layers of a system	166
SDM types	167
SDM schema structure	169
Definition complex type	170
ObjectDefinition complex type	172
RelationshipDefinition complex type	175
Member complex type	176
Settings	179
Constraints	182
SDM documents	186
Prototypes and the Distributed System Designers	189
The SDM SDK	190
he SDM command-line compiler	191
Walkthrough: SDM Sample	192
SystemDefinitions	193
Settings	195
Endpoints	196
Communication and delegation	200
Containment	201
Hosting	206
Flows	207
Constraints	208
Creating your own constraints	211
SdmC	216
ProtoGen	216
The registry	217
Visual Studio	218
Summary	219
Part Two: Team Developer	221
Chapter 8: Managed Code Analysis	223
The Need for Analysis Tools	223
Using Managed Code Analysis	224
Built-in Managed Code Analysis rules	225
Enabling Managed Code Analysis	226
Executing static code analysis	227
Working with rule violations	230

Contents

Using the Command-Line Analysis Tool	233
FxCopCmd options	233
FxCopCmd project files	235
Build process code analysis integration	236
Creating Code Analysis Rules	237
Reflection and Introspection	237
Creating a new rule	237
Summary	244
Chapter 9: Code Analysis for C/C++	245
Understanding Static Code Analysis	246
How the C/C++ Code Analyzer Works	247
Visual Studio 2005 Integration	248
Enabling and disabling C/C++ Code Analysis	248
Setting warning levels in Visual Studio 2005	249
Viewing code analysis warnings and errors	250
Command-line Support	251
Annotation Support	253
Working with annotations in C++	254
Working with annotations in C	254
#pragma Support	255
Integrating with Team Foundation Server	257
Creating check-in policies using C/C++ code analysis	257
Setting check-in notes for your C/C++ code	259
Creating work items from code analysis warnings	260
Identifying and Minimizing Noise	261
Tool Limitations	262
Summary	262
Chapter 10: Application Verifier	263
Setup and Configuration	264
Troubleshooting Heaps	266
Troubleshooting Handles	270
Troubleshooting Locks	271
Application Verifier End-To-End	274
Programming Application Verifier	278
Summary	281

Chapter 11: Refactoring and Code Snippets	283
Refactoring from Class Diagrams	284
Extract Interface	284
Implement Abstract Class	287
Refactoring in Code	288
Rename	288
Encapsulate field	291
Extract Method	294
Promote local variable to parameter	296
Reorder parameters	297
Remove parameters	298
Generate Method Stub	299
Improving Code Quality with Code Snippets	300
Using code snippets in Visual Basic	300
Using code snippets in Visual C#	302
Code Snippets Manager	303
Summary	305
Chapter 12: Profiling and Performance	307
Introduction to Performance Analysis	307
Types of profilers	308
Team System profiling	308
Using the Team Developer Profiler	309
Creating a sample application	309
Creating a performance session	311
Performance Explorer	313
Configuring a sampling session	318
Configuring an instrumentation session	319
Executing a performance session	320
Managing session reports	321
Reading and interpreting session reports	321
Command-line Execution	330
Configuring instrumentation	330
Setting environment variables	331
Executing a profiling session	332
Generating reports	332
Modifying Instrumentation Programmatically	333
Common Profiling Issues	335
Debugging symbols	335
Instrumentation and code coverage	335

Contents

Virtual machines	336
ASP.NET profiling	336
Summary	336
Part Three: Team Tester	337
Chapter 13: Test Case Management	339
Test Projects	339
Creating a test project	340
Setting test project options	341
Managing Test Cases	342
Test Manager	342
Test View	348
Working with Test Results	349
Test Results window	349
Sorting, grouping, and filtering tests results	350
Exporting test results	350
Using Test Results XML (TRX) files	350
Publishing results	351
Using Ordered Tests	352
Creating an ordered test	352
Ordered test properties	353
Executing and analyzing ordered tests	354
Summary	355
Chapter 14: Unit Testing with the Unit Test Framework	357
Unit Testing Concepts	358
Benefits of unit testing	358
Writing effective unit tests	359
Test-driven development	359
Third-party tools	361
Team System Unit Testing	361
Creating your first unit test	362
Managing and running unit tests	364
Test run configuration	366
Test results	367
Debugging unit tests	368
Programming with the Unit Test Framework	368
Initialization and cleanup of unit tests	368
Using the Assert methods	371
Using the CollectionAssert class	374

Using the StringAssert class	375
Expecting exceptions	376
Defining custom unit test properties	377
TestContext class	377
Creating data-driven unit tests	378
Accessing Nonpublic Members from Tests	379
Using PrivateObject to access nonpublic instance members	379
Using PrivateType to access nonpublic static members	381
Code Generation	382
Generating tests from code	382
Generating code from tests	386
Unit Testing ASP.NET Applications	387
ASP.NET unit test attributes	387
Creating ASP.NET unit tests	388
Code Coverage	389
Enabling code coverage	390
Viewing code coverage results	390
Summary	392
Chapter 15: Web and Load Testing	393
Web Tests	394
Creating a sample web application	394
Creating and configuring web tests	396
Data-driven web tests	407
Coded web tests	409
Load Tests	411
Creating and configuring load tests	411
Editing load tests	419
Executing load tests	422
Viewing and interpreting load test results	422
Command-line Test Execution	426
Executing tests	426
Executing test lists	426
Other test options	427
Distributed Load Tests	427
Installing controllers and agents	427
Configuring controllers	428
Configuring agents	428
Test run configuration	429
Running a distributed load test	430
Viewing a distributed load test	430
Summary	430

Chapter 16: Manual Testing	433
Test Automation versus Manual Testing	433
Test automation	434
Manual testing	435
Test Cases	437
Planning Your Manual Tests	437
Manual Testing in Team System	438
Creating manual tests	439
Configuring manual test options	440
Manual test templates	442
Manual test structure	448
Managing Manual Tests	449
Using the Test Manager	450
Logging your manual test as a bug work item	451
Publishing the test results on Team Foundation Server	452
Summary	453
Chapter 17: Generic Testing	455
Creating and Running a Generic Test	456
Creating a generic test	456
Running the generic test	462
Creating an External Testing Tool	464
Scripting Host example	464
Managed Code example	465
Extended Return Results	467
Summary report XML schema	468
Example summary report XML file	469
Wiring to Existing Test Systems	470
Creating a wire to NUnit	471
JScript conversion code	473
Summary	477
Part Four: Team Foundation	479
Chapter 18: Team Foundation Architecture	481
What Is Team Foundation Server?	482
Team processes	482
Microsoft Solutions Framework (MSF)	482
Project management guidance	483
Team Foundation features	483

Version control	484
Work item tracking	487
Team Project Portal	488
Team Foundation Build	488
Team Foundation Reporting	488
Team Foundation core services	489
Team Edition Test Load Agent	489
Team Foundation Architecture	489
Clients of Team Foundation	490
Team Explorer	490
Team Foundation Build	494
Team Foundation Reporting Warehouse architecture	494
Configuration	496
Summary	497
Chapter 19: Project Management Tools and Team Reporting	499
Creating a Software Project	500
Why software development projects fail	501
Using the New Team Project Wizard	502
Administering Project Details	505
Connecting to the Team Foundation Server	505
Configuring Project Settings	507
Project Alerts	511
Team Portal customization and extensibility	513
Deleting a Team Project	514
Work Item Tracking	516
Anatomy of a Work Item	516
Creating Work Items Using Visual Studio	517
Creating and running work item queries	518
Creating and assigning work items using Microsoft Office	519
Creating and customizing work item types	522
Team Foundation Server Reporting	524
Working with Team Reports	524
Viewing reports using Microsoft Excel pivot tables	529
Building custom reports	530
Report site security	533
Summary	534
Chapter 20: Team Foundation Version Control	535
Comparing Team Foundation Source Control and Visual SourceSafe (VSS) 2005	536
Source code migration	540

Contents

Setting Up Version Control	541
Setting up security roles	542
Setting up your workspace	543
Using the Source Control Explorer	544
Workspaces	546
Adding Projects to the Source Repository	548
Check-in and Check-out	549
Checking in an item	549
Checking out an Item	550
Creating and administering check-in policies	551
Branching and Merging	553
Branching	554
Merging	555
Shelving	557
Command-Line Tools	560
Summary	561
Chapter 21: Microsoft Solutions Framework	563
About the Microsoft Solutions Framework	564
A conceptual overview of MSF	565
Obtaining the Microsoft Solutions Framework	566
MSF for Agile Software Development	566
About Agile development	568
MSF for CMMI Process Improvement	569
Capability Level 3	571
CMMI Level 3 vs. MSF for CMMI process improvement	571
Choosing the Right Process	573
Fundamental Concepts	574
Process template components	574
Iterative process	575
Cycles	575
Personas	576
Governance, tracks, and activities	577
Work item types	578
Reports	582
Team Model	582
Advocacy	584
Mindsets	585
Roles	585
Business analyst	586
Project manager	587
Architect	587

Developer	588
Tester	588
Release manager	589
User experience	589
Workstreams	589
Summary	590
<hr/>	
Chapter 22: Process Templates	591
<hr/>	
New Team Project Wizard	592
Customization Tools	592
XML/XSL editors	593
Microsoft InfoPath 2003	593
MSF HTML Build utility	594
Third-party tools	594
Customizing Process Guidance	595
Anatomy of process guidance (ProcessGuidance.xml)	595
Creating workstreams and activities	596
Creating work item documentation	599
Creating custom roles	600
Cross-references	600
Viewing your custom guidance	601
Customizing Process Templates	601
Downloading a process template to your desktop	601
Navigating a process template	602
Modifying template definitions	603
Uploading process templates in Team Foundation Server	622
Deleting process templates	623
Testing Your Custom Process Template	623
Summary	623
<hr/>	
Chapter 23: Team Foundation Build	625
<hr/>	
Setting Up Team Foundation Build	627
Planning a build strategy	628
Setting up your own build lab	629
Team System integration	629
Setting up build security	630
Using Team Foundation Build	630
Creating build types	630
E-mail notifications	635
Executing a local build	636
Getting the Team Foundation Build status	637

Contents

Command-line interface	638
Starting a build	638
Deleting a build	638
Stopping a build	639
Setting up a nightly build	639
Continuous integration (CI)	640
Team Foundation Build Reporting	641
Build Customization and Extensibility	644
Creating MSBuild tasks	645
Summary	646
Chapter 24: Deployment and Administration	647
Setting Up Your Operations Team	648
Planning a Deployment	649
Unsupported deployment scenarios	650
Capacity planning	650
User accounts	651
Firewall configuration	652
Network topologies	652
Creating a test plan	654
Test rig considerations	655
Considerations for Migrating and Integrating Your Existing Tools	655
Version control	656
Work-item tracking	657
Reporting	657
Testing tools	657
Backup and Recovery	658
Licensing Models	659
Where to get Team System	659
Summary	660
Index	661