



Index

A

- AbsoluteBounds setting, 413, 415, 449
- Abstract base roles, multiplicity of, 127
- Abstract classes, 67–68, 90, 93, 95, 117
- Abstract relationships, 107, 115–116, 245
- Abstraction
 - in domain models, 462
 - in language evolution, 12, 17
 - in UML, 39
- Accept code
 - connection builders, 223–226
 - custom, 208–210
- Access modifiers, 116, 120, 123
- Accessing links, 124–126
- Action setting, 105, 243–244, 250
- Active Server Pages, 320
- Add Custom Editor option, 137, 142
- Add Designer option, 137
- Add Diagram option, 137
- Add Domain Path option, 187
- Add ExplorerNodeSettings option, 184
- Add Exposed option, 139
- Add menu in explorer, 200, 203–205
- Add menu option, 435
- Add method, 215
- Add New option, 218
- Add New Domain Enumeration option, 96
- Add New Domain Property option, 95
- Add New Domain Type Descriptor option, 192
- Add New Item option
 - for deployment, 76
 - for files, 66
 - for templates, 383, 393–395
- Add New Project option, 373
- Add New Property Path option, 186
- AddButton_Click method, 145
- AddElement method, 145
- AddGraph method, 215, 437
- Additions in DSL evolution, 490
- AddObserver method, 293
- AddRule class, 414
- Allows Duplicates setting, 106, 249
- AlwaysRequired flag, 332
- Analysis model
 - properties for, 280
 - in UML, 35–36
- APIs
 - for artifacts, 314–319
 - for framework driving, 484
- app.config file, 29–30
- AppDomain, 356–359
- Appearance
 - diagrams, 138
 - explorer, 181–183
 - nodes, 184–186
 - properties window, 188–190
 - shapes, 147, 151
- AppendLine method, 366
- Application Designer, 27–31
- Application users, 54
- Applications
 - generalizing, 42–46
 - generative, 74–75
 - interpretive, 75–76

Architecture

- DSL tools, 78
 - Dsl project contents, 81–83
 - DslPackage project contents, 83–85
 - framework assemblies, 79–81
 - generated code, 78–79
 - layers, 79–80
- interpretive, 56
- text templating, 352–354

Arrows and arrowheads, 110–112, 411

Artifacts, 24–25

- complex relationships and round-tripping
 - for, 321–324
- customizations, 351–352
- domain-specific APIs for, 314–319
- extensible stylesheet language
 - transformations for, 311–314
- large-scale, real-world, 349–351
- overview, 309–310
- template-based approach. *See* Templates and templatization process

Assemblies

- compiled, 77
- framework, 79–81

Assembly directive, 343

AssemblyCache class, 357

AssemblyInfo.cs file, 82, 304–305

assemblyPath attribute, 380–381

Assertions, 275

AssociationConnector class, 409–410

Authoring solutions in DSL development, 57–60

Authors, 54–55

AutoLayoutShapeElements method, 180

Automating tests, 485

B

Background pictures in diagrams, 176–177

Backus Naur Form (BNF), 15–16

Base classes, abstract, 67

Base roles, multiplicity of, 127

BeginErrorSession method, 362

BeginTransaction method, 114

Behaviors

- explorer, 182–183
- in graphical DSLs, 22–23
- inheriting, 292

Bentley, Jon, 13

Bin directory, 374

BinaryLinkShape class, 413

BNF (Backus Naur Form), 15–16

Booch, Grady, 34

Booch method, 34

Boolean type

- for constraints, 276
- for properties, 96, 119

Bottom-up approach

- in incremental development, 46–48
- in variability identification, 457

Bounds rules, 420–424, 448–449

BoundsRules setting, 421–422

Brackets (<# and #>) in code-generation statements, 52

Bring Tree Here option, 94

Broken diagram files, 256–257

Build Solution option, 144

Building menu commands, 435

Business Class Model, 8

C

C# language and classes, 92

- for constraints, 281
- in domain models, 110
- in templates, 331

Calculated properties, 121

- for change, 404–405
- for dependent values, 407
- rules with, 121, 416

callback parameter, 362

Camel-casing, 313

CanAcceptComponentAndComponentAsSourceAndTarget method, 225

CanAcceptSource method, 228

CanAcceptSourceAndTarget method, 228

CanAcceptXxxAsSource methods, 224

CanAcceptXxxAsSourceAndTarget methods, 224

CanAcceptXxxAsTarget methods, 225

CanCreateConnection method, 302

Candidate notations, 495–497

CanFocus method, 401

CanManuallyRoute method, 402

CanMerge method, 401

CanMergeIssueCategory method, 208–209

CanMove method, 401

CanSelect method, 401

Captions for tools, 65, 199

Catalysis approach, 460

- Categories
 - properties window, 190–191
 - shapes, 137, 147
 - validation, 289–292
- Category setting, 120, 287
- Category tool, 202
- CategoryTreeConnector class, 180
- Change, 402
 - bounds rules, 420–424
 - calculated properties for, 404–405
 - custom storage domain properties for, 405–407
 - in domain models, 462
 - in DSL evolution, 489–490
 - events for, 417–420
 - NotifyValueChange for, 407–408
 - OnAssociatedPropertyChanged for, 408–411
 - property handlers for, 402–404
 - rules for, 412–417, 449–451
 - summary, 424–426
 - undoable, 424
- ChangeRule class, 414–415
- CheckVersion method, 266
- Child shapes, 442
 - moving, 451–453
 - nested, 443–446
- Circuit Diagrams DSL
 - delete propagation in, 231–234
 - image shape in, 174–176
- CircuitsCommandSet class, 438
- CircuitsToolboxHelper class, 214–215
- CJKWCustomEditor class, 144–146
- CJKWCustomEditorDocView class, 143
- Class Data node, 259, 261–262, 271
- Class Designer, 27, 110, 324
- Class diagrams, 24–25
 - generated domain classes, 111–113
 - menu commands for, 429–435
 - UML, 34–35
- Class feature blocks, 331–332, 347–348
- Class members in UML, 37
- ClassDiagramsDomainModel class, 298
- Classes, 92
 - creating, 92–98
 - deleting, 208, 490
 - in domain models, 48–50
 - double derived, 398–399
 - element tools for, 199
 - obsolete, 491
 - partial, 116–117, 397–399
- Classes language template, 58
- ClassStyleSet method, 402
- ClearIndent method, 348
- ClinicalSignoffReceived method,
 - 336, 338, 340
- ClinicalSignoffRequired method, 336, 338
- CLR (Common Language Runtime), 37, 120
- CLRAAttributes setting, 285
- Code coverage measurements, 485–486
- Code Coverage Results option, 485–486
- Code customization options, 115
- Code extensions, 33
- Code generation, 60, 78–79
 - in DSL development, 66–67
 - for framework driving, 484
 - serialization, 264–270
 - statements for, 51–53
 - testing, 488–489
- Code settings
 - diagrams, 138–139
 - shapes, 147
- CodeDOM facility, 318–319, 366
- Collapse method, 445
- Collection Type setting, 123
- Color
 - decorator text, 168
 - shape text, 149
 - shapes and connectors, 136, 411
- Combinatorial tests, 485
- Commands
 - building and running, 435
 - command handlers for, 432–435
 - command sets for, 431–432
 - custom, 484
 - ids for, 430
 - resource indexes for, 430–431
- Commands.ctc file, 84
- CommandSet class, 291, 428
- CommandSet.cs file, 84
- Comment class, 99
- Comment Link tool, 223
- CommentBoxShape class, 173–174
- CommentReferencesStateElements
 - relationship, 99
- CommentReferencesSubjects
 - relationship, 217
- CommentReferToIssueStates relationship,
 - 99–100
- Comments, associating with projects,
 - 205–207

- <comments> element, 241–243
- CommentsReferToIssueStates relationship, 112–113, 241
- <commentText> element, 241
- Commit method, 440
- Common Language Runtime (CLR), 37, 120
- Compartment shapes
 - decorator positions for, 170
 - and element merge directives, 202–203
 - mapping, 160–161
 - overview, 149–151
- Compartments, relationship notation for, 469
- Compiled assemblies, 77
- Compiler for custom code, 144
- Complex relationships, 321–324
- ComponentDiagram class, 177
- ComponentHasComponentTerminal
 - option, 231
- ComponentHasPorts relationship, 230
- ComponentImageField class, 175
- Components language template, 58
- ComponentsDeleteClosure class, 233
- ComponentTerminal domain class, 231
- Configuration code, 17
- Configuration files for framework
 - driving, 483
- Connect method, 228–229
- Connection builders, 199–200, 216–217
 - accept code for, 223–226
 - connect code for, 226–229
 - multiple link connect directives for, 219–221
 - multiple source and target role directives for, 217–219
 - validating, 222
- Connection points, 177–178
- Connection tools, 198
- ConnectionBuilder class, 228–229
- ConnectionBuilders.cs file, 83
- Connections, 20
- Connectors
 - anatomy, 164–165
 - in domain models, 49–50
 - inheritance, 94, 165
 - maps, 165–167
 - per-class relationships involving, 410
 - properties, 136, 189–191
 - relationship notation for, 469
 - routing style, 165, 178–180
- Connectors.cs file, 83
- Consistency checks
 - external, 478
 - internal, 476–478
- Constants.cs file, 84
- Constraints, 24, 51, 68–70, 226, 295–296. *See also* Validation
 - custom accept, 226
 - defining, 475–478
 - delete propagation, 231
 - for external consistency, 478
 - hard vs. soft, 277–280
 - hard with soft, 299–307
 - for internal consistency, 476–478
 - overview, 275–277
 - rules for, 296–298
 - summary, 424–426
 - testing, 486–487
- Constructors, custom, 399
- Consumer endpoints, 27
- ContainerControl class, 436
- Containers, 442
 - child shapes, 442–446
 - in issue state notation, 471
 - rules for, 446–453
- ContainerShapeChangesRule
 - class, 412, 449
- ContainerShapeInlineChangesRule class, 452
- Context, transaction, 416–417
- ContextInfo class, 417
- Control blocks, 330–331
 - class feature, 347–348
 - expression, 349
 - languages in, 342
 - standard, 346–347
- Controller classes, 34
- Copy and paste, 437
 - Copy method, 437–438
 - paste method, 438–440
 - registering menu handlers, 440–441
- Copy method, 437–438
- Coverage analysis
 - code, 485–486
 - text templates, 488
- CreateElementToolPrototype method, 214, 402
- CreateIssue method, 316–317, 319
- CreateNewHealthIssue method, 326
- Cross-referencing in serialization, 245–251
 - .cs extension, 51

CSharpCodeProvider class, 281
 CtcComponents, 431
 Culture issues in graphical notation, 140–141
 Culture parameter in template directive, 342
 CurrentData setting, 291
 CurrentIndent method, 348
 Custom accept option, 222–226
 Custom Attributes setting, 117, 193
 Custom code, 73
 Custom commands, 484
 Custom connect option, 223, 226–229
 Custom directives, 344–346
 Custom editors, 142–146
 Custom hosting, 354–357
 Custom Node Settings, 185
 Custom Post Load setting, 259
 Custom storage property rules, 416
 Custom Tool setting, 332, 359
 CustomAccept setting, 400
 CustomCategories setting, 287
 CustomConnect setting, 400
 customFragmentIds attribute, 380
 Customization
 constructors, 399
 customization pit, 32–34
 delete propagation, 232–234
 explorer, 187–188
 in good DSLs, 492
 overrides, 233, 401–402
 overview, 397
 partial classes for, 397–399
 properties window, 72–73, 193
 responding to change. *See* Change
 serialization, 258–264
 setup, 377–378
 switches, 399–400
 CustomMonikerizeReference
 method, 264
 CustomPostLoad setting, 400
 CustomRead method, 271
 CustomSerializeReference method, 264
 CustomStorage setting, 121, 405–407
 CustomWrite method, 271
 CustomWriteElements method, 272
 CustomWritePropertiesAsElements
 method, 271–272
 Cut method, 438
 CyclicInheritanceAddRule class, 48, 297–299

D

Data access layer (DAL) generator, 289
 Data Contract DSL, 6
 Data loss in complex relationships, 321
 Data structures in templates, 350
 DatabaseName setting, 294
 Debug code in temp directory, 342
 debug parameter, 342
 Debugging project
 files in, 62–63, 66
 text templates from, 390–392
 Decorators, 167
 kinds, 168–169
 maps, 170–172
 multiline, 173–174
 positioning, 169–170
 Default Expand Collapse State setting, 150
 Default Text decorator resource, 168
 defaultDirectoryName attribute, 380
 DefaultGridSize method, 402
 DefaultRoutingStyle method, 402
 Defaults
 delete propagation rules, 229–230
 explorer appearance, 181–183
 names, 101
 properties window appearance, 188–190
 DefaultWriteElements method, 272
 Definition properties category, 115
 Definition settings
 diagrams, 139
 shapes, 147
 Definitional elements, 246
 Delete Behavior tab, 229–230
 Delete method, 124
 Delete propagation
 controlling, 231–232
 customizing, 232–234
 default rules, 229–230
 DeleteRule class, 414
 Deleting
 diagrams, 239
 domain classes, 208, 491
 in DSL evolution, 490
 elements, 229–234
 DeletingRule class, 414
 Delimiters in templates, 330
 Deployment, 76–77
 .dslsetup for. *See* .dslsetup file
 files required, 370–373

- Deployment (*Continued*)
 - overview, 369–370
 - package load keys, 388–390
 - refreshing installation files, 387
 - setup projects for
 - contents, 376–377
 - creating, 373–376
 - customizing, 377–378
 - text templates, 390–395
- Derivation, relationship, 106–108, 126–128, 245
- Description setting, 98
 - for domain classes, 117–118
 - IssueState, 95–96, 268
 - for properties, 121
- descriptionKey attribute, 381
- Descriptions in properties window, 190–191
- Design time, 54
- Designer editors, 142
- Designing DSLs
 - bottom-up and top-down approach, 457
 - domain models, 460–468
 - evaluation in, 491–498
 - evolution in, 489–491
 - feature trees for, 458–460
 - frameworks in, 479–484
 - notation in, 468–469
 - familiar, 474–475
 - issue state, 471–474
 - project definition, 470–471
 - overview, 455–456
 - testing in, 484–489
 - validation constraints in, 475–478
 - variability in, 456–457
- DesignSurfaceElementOperations class, 439
- Device interfaces, 9
- Diagram class, 142
- Diagram.cs file, 83
- Diagram Element Map tool, 155
- .diagram extension, 238
- DiagramDocView class, 436
- Diagrams
 - background pictures in, 176–177
 - class, 24–25
 - generated domain classes, 111–113
 - menu commands for, 429–435
 - UML, 34–35
 - definitions, 138–140
 - deleting, 239
 - embedding in controls, 435–437
 - fixing, 256–257
 - saving and loading, 238
 - serialization, 251–254
- Dictionary class, 283
- Digital media in language evolution, 12
- Dimitriev, Sergey, 13
- Direction of connectors, 165
- Directive processors, 344–345, 357–359, 386
- DirectiveProcessor.cs file, 83
- Directives
 - connection builder, 217–221
 - merge. *See* Element merge directives (EMDs)
 - template, 329–330, 340–346
- Disallowed links, 462
- Display decorator setting, 170
- Display Name setting, 98, 117, 120
- Displayed elements collection path setting, 161
- Distributed System Designers, 27
- DocData class, 264–265, 304, 417, 428
- DocData.cs file, 85, 264–265
- Documentation settings
 - diagrams, 139
 - shapes, 147
- DocView class, 429, 436
- DocView.cs file, 85
- Domain Class tool, 92–93
- Domain classes. *See* Classes
- Domain models, 87–88
 - classes in, 92–98, 115–118
 - in development of DSLs, 48–51
 - Domain Model Designer for, 88–89
 - generated code for, 109–115
 - in graphical DSLs, 24
 - inheritance in, 93–95
 - properties in, 95–98, 119–122
 - relationships in, 98–101, 122–124
 - derivation, 106–108, 126–128
 - embeddings, 101–103
 - link access, 124–126
 - multiplicities in, 100–104
 - reference, 105–106
 - roles in, 122–129
 - without shapes, 108–109
- snapshots, 460–461
 - issue state, 462–464
 - project and field, 461–462
 - project definitions, 464–468

- Store for, 89–92, 129–131
 - validating, 89
 - in Visual Studio, 64
 - Domain paths, 187
 - Domain properties. *See* Properties
 - Domain relationships. *See* Relationships
 - Domain roles. *See* Roles
 - Domain-specific APIs for artifacts, 314–319
 - Domain-Specific Development (DSD)
 - overview
 - benefits, 10–11
 - device interfaces, 9
 - embedded systems, 9
 - examples, 4–8
 - overview, 1–4
 - related work, 13–15
 - software defined circuitry, 8–9
 - software development process
 - customization, 9
 - Domain-Specific Language (DSL) overview
 - customization pit in, 32–34
 - definition, 11–12
 - graphical. *See* Graphical DSLs
 - textual, 15–20
 - UML, 34–40
 - in Visual Studio, 27–31
 - Domain-Specific Language Setup template, 77, 373
 - Domain-Specific Modeling, 14
 - DomainClasses.cs file, 82, 209
 - DomainClasses.tt template, 209
 - DomainClassInfo class, 118, 129
 - DomainModel.cs file, 83
 - DomainModelInfo class, 130–131
 - DomainModelResx.resx file, 83
 - DomainProperty setting, 400
 - DomainPropertyInfo class, 121–122
 - DomainRelationshipInfo class, 129
 - DomainRelationships.cs file, 82
 - DomainRoleInfo class, 124, 129
 - DomainRoles setting, 129
 - Double arrowheads, 110–112
 - Double-derived patterns, 265, 398–399
 - Driving
 - applications, 74–76
 - frameworks, 51–53, 483–484
 - DSL author role, 54–55
 - Dsl.csproj file, 82
 - DSL design time, 54
 - DSL Designer Wizard, 57–59
 - DSL Details window
 - for compartment mapping, 160
 - for delete propagation, 231
 - visibility of, 155
 - Dsl project, 60–61, 78, 373
 - contents, 81–83
 - DslPackage as friend of, 304–305
 - DSL runtime, 54
 - DSL Tools Redistributable, 372, 374–376
 - DSL user role, 54–55
 - DslDefinition.dsl file, 64, 82
 - DslDefinition.dsl.diagram file, 82
 - DSLExplorer, tools in, 198
 - <dslPackage> element, 380–381
 - DslPackage project, 60–61, 78, 373
 - contents, 83–85
 - as friend of Dsl, 304–305
 - <dslSchemas> element, 384
 - .dslsetup file, 376–380
 - definitions for, 385–387
 - <dslPackage> element, 380–381
 - <dslSchemas> element, 384
 - <licenseAgreement> element, 382
 - <mergeModules> element, 384–385
 - <supportingFiles> element, 382–383
 - for text templates, 391–393
 - <textTemplates> elements, 385
 - <vsItemTemplates> element, 383–384
 - <vsProjectTemplates> element, 384
 - DslTextTransform.cmd script, 355
 - DSLToolsRedist.msi file, 372, 374–376
- ## E
- edgePoints attribute, 254
 - Editing labels, 65
 - EditorFactory class, 141, 428
 - EditorFactory.cs file, 85
 - Editors
 - for graphical notation, 141–142
 - custom, 142–146
 - designer, 142
 - for templates, 320
 - Efficient validation code, 284
 - Element Data node, 261–264
 - Element group prototypes (EGPs), 437
 - Element merge directives (EMDs), 200–202
 - Add menu for, 203–205
 - and compartment shapes, 202–203
 - custom, 208–211
 - forwarding, 207–208

- Element merge directives (EMDs) (*Continued*)
 - multiple, 202
 - multiple link creation paths in, 205–207
 - re-parenting with, 211–212
 - Element Name setting, 259
 - Element Name Provider setting, 120
 - Element-to-link path segments, 158–159
 - ElementAdded method, 297–298
 - ElementGroups, 215
 - ElementLink class, 90–91, 413
 - ElementNameProvider class, 120
 - ElementPropertyChanged method, 412, 415, 449, 452
 - Elements
 - connection builders. *See* Connection builders
 - creating, 62, 197–198
 - deleting, 229–234
 - EMDs for. *See* Element merge directives (EMDs)
 - links, 90–91
 - names, 259–261
 - serialization, 242
 - Store, 90–91
 - tools for, 198–200, 212–216
 - XML. *See* XML (Extensible Markup Language)
 - embed attribute, 382
 - Embedded DSL, 17
 - Embedded systems, 9
 - Embedding links, 105, 124, 197, 201–202, 207
 - Embedding Relationship tool, 102
 - Embedding relationships, 49, 101–103, 124
 - element merge directives for, 200–201
 - explorer behavior for, 182
 - serialization, 242
 - EMDs. *See* Element merge directives (EMDs)
 - End-to-end tests, 485
 - EndErrorSession method, 362–363
 - Endpoints, 20, 27, 446–447
 - Engines, text templating, 352–353
 - EnsureConnectionPoints method, 178
 - Entity classes, 34
 - Enumerations, 96, 119
 - Error method, 347
 - ErrorListObserver class, 293–294
 - EULA (license agreements), 377
 - Events
 - for model changes, 322
 - .NET, 419–420
 - overrides, 420
 - for stores, 121, 417–419
 - Evolution
 - DSLs, 55–56, 489–491
 - frameworks, 482–483
 - languages, 12, 17
 - ExampleChild class, 443
 - ExampleElement class, 443
 - ExampleShape class, 445
 - Expand collapse decorators, 168–169
 - Expand method, 445
 - Experimental hive, 64
 - Expert plateau, 34
 - Explorer, 63, 180
 - Add menu, 203–205
 - appearance, 181–183
 - customizing, 71–72, 187–188
 - nodes in, 184–188
 - relationship notation for, 469
 - window icons and labels for, 183–184
 - Exposed Style Properties settings, 139, 147
 - Exposes XXX As Property setting, 139
 - Exposing style properties, 139
 - Expression blocks, 331, 349
 - Expressions, regular, 1, 493–494
 - Extensible code, 33
 - Extensible Markup Language. *See* XML (Extensible Markup Language)
 - Extensible stylesheet language
 - transformations, 311–314
 - extension attribute, 381
 - Extension facilities in UML, 37
 - Extensions, file, 63–64
 - External APIs for framework driving, 484
 - External data validation checks, 294–295, 478
- ## F
- Factories, 6–8, 14–15
 - Familiarity in good DSLs, 492
 - Families of languages, 13
 - Feature blocks, 331–332, 347–348
 - Feature trees, 458–460
 - Field Programmable Gate Arrays (FPGAs), 8
 - Field snapshots, 461–462
 - FieldType enumeration, 331
 - FieldTypeToType method, 329, 332, 335, 341
 - <fileExtension> element, 381
 - filePath attribute, 382

- Files
 - extensions, 63–64
 - included, 79, 340–341, 344, 392–393
 - migrating, 314
 - FillGradient setting, 398
 - Filmstrips, 463–464
 - FindElement method, 129
 - fixedFrom attribute, 254
 - fixedTo attribute, 254
 - FixUpDiagram class, 444
 - Flag properties, 119
 - Flexibility in generated vs. interpretive frameworks, 480
 - Flip setting, 175
 - Flow links, 218–221
 - FlowElements
 - state charts, 446–447
 - Task Flow, 212, 218
 - Folders, namespaces in, 211
 - Forward merge option, 207
 - Forwarding
 - element merge directives, 207–208
 - properties, 189, 192–193
 - Fowler, Martin, 14
 - FPGAs (Field Programmable Gate Arrays), 8
 - Frameworks
 - driving from DSLs, 51–53, 483–484
 - evolving, 482–483
 - generation vs. interpretation, 479–482
 - in tool architecture, 79–81
 - Full-form qualified names, 248
 - Full-form serialization, 244
- G**
- GenClass directive, 364
 - Generalization class, 305
 - Generalization link directive, 225
 - GeneralizationConnectAction class, 301–302
 - GeneralizationConnectActionType class, 301–302
 - Generalizing applications, 42–46
 - GenerateCreateIssue method, 317
 - Generated code
 - for domain models, 109–115
 - serialization, 264–270
 - testing, 488
 - in tool architecture, 78–79
 - GeneratedCode folder, 61, 66, 78–79
 - GeneratedCommand.h file, 84
 - GenerateEmptyClass method, 348
 - GeneratePostInitializationCode method, 365–366
 - Generates Double Derived setting, 117, 173, 177, 266, 398, 410
 - GenerateTransformCode method, 364–365
 - Generation
 - frameworks, 479–482
 - graphical DSLs, 24–25
 - Generation Gap design pattern, 398–399
 - Generative applications, 74–75
 - Generative Programming, 14
 - Generative Programming and Component Engineering (GPCE) conference, 14
 - Generator template testing, 488
 - Generic-specific model data in templization process, 338–340
 - Geometry shapes, 137
 - decorator positions for, 170
 - mapping, 155
 - overview, 148–149
 - GetCompliantBounds method, 422
 - GetCustomDomainModelTypes method, 130, 298
 - GetDisplayImage method, 175
 - GetElement method, 129
 - GetFieldType method, 335
 - GetHostOption method, 355
 - GetLinks method, 125
 - GetMenuCommands method, 431, 440–441
 - GetNameAndTypeValue method, 405
 - GetPrice method, 29, 31
 - Getter Access Modifier setting, 120
 - GetValue method, 122
 - GetXXXValue methods, 121, 405
 - Globally Unique Identifiers (Guids)
 - command groups, 440
 - for commands, 435
 - ModelElement, 129
 - package, 389
 - as references, 246–247
 - Graphical DSLs
 - behavior representation, 22–23
 - domain models, 24
 - generation, 24–25
 - notation. *See* Notation
 - overview, 20–21
 - serialization, 25
 - structure representation, 21–22
 - tool integration, 26
 - Graphical languages, 2–3, 11–12

Guidance Automation Toolkit, 484
Guids. *See* Globally Unique Identifiers (Guids)

H

HandleAbsoluteBoundsChange method,
450, 452
Handlers
command, 432–435
menu, 440–441
property, 402–404
Hard constraints, 226, 295–296
rules for, 296–298
vs. soft, 277–280
with soft, 299–307
summary, 424–426
Hardware Description Language (HDL), 8
Has Custom Constructor setting, 117, 399
Has Custom Moniker setting, 264, 270
Has Custom Parent Element setting, 156, 444
Has custom parent shape setting, 157, 163
Has custom source setting, 166
Has custom target setting, 166
Has Default Connection Points setting,
152, 178
HasBackgroundGradient method, 398
HasConnectionPoints method, 178
HasCustomMoniker switch, 400
HasCustomParentElement switch, 400
HasCustomXxx flags, 173
hasIcon attribute, 381
HasShadow method, 401
HDL (Hardware Description Language), 8
HealthcareProjectIssue class, 336–337
HealthIssue class, 327–328
Height
compartment shapes, 150
image shapes, 152
swimlanes, 154
Help keywords, 199
Hiding
nodes, 186–188
properties, 192
hierarchy parameter, 362
Himalia Navigation Model, 4–5
Horizontal DSLs, 8
Horizontal swimlanes, 154
Hosts
graphical DSLs, 20
text templating, 352
hostspecific parameter, 342

I

Icon decorators, 168–169, 171
Icon setting, 95–96, 268
iconId attribute, 381
Icons
explorer, 183–184
resources for, 152
for tools, 199
Ids and Id setting
DomainInfo, 118
Guids, 247
for menu commands, 430
ModelElement, 129
Image Resource setting, 151
Image shapes, 137, 151–152
decorator positions for, 170
resources for, 152
variable, 174–176
Images, resources for, 152
Implement All Applications option, 28
Implementing serializers, 264
Import directive, 343
Important bit emphasis in good DSLs, 492
In-memory Store. *See* Store
Included files, 79, 340–341, 344, 392–393
Incremental development of DSLs, 41–42
domain models in, 48–51
evolution in, 55–56
framework driving in, 51–53
generalizing applications in, 42–46
interpretive architectures, 56
top-down and bottom-up, 46–48
using DSLs in, 53–55
Index numbers for commands, 435
Inheritance
connectors, 94, 165
in domain models, 93–95
looped, 299–307
relationships, 466–468
shape, 154
validation behavior, 292
Inheritance setting, 67
Inheritance Modifier setting, 107, 115
Inheritance tool, 93
inherits parameter, 342
Initial Height setting
compartment shapes, 150
image shapes, 152
swimlanes, 154
Initial Width setting, 152

- Initialize method
 - package declarations, 427–428
 - text templates, 353–354, 358
- InitializeInstanceResources method, 419
- InitializeResources method, 409–410
- InitializeShapeFields method
 - ComponentDiagram, 177
 - TransistorShape, 176
- InitiallyDisabled setting, 414
- Inline rules, 451
- InnerStateTransition relationship, 127
- InPortBoundsRules class, 423
- inputFile parameter, 362
- Installation files, refreshing, 387
- Installation wizard, 375–376
- InstallerDefinition.dslsetup file, 377
- <installerDefinition> element, 379–380
- installShortcut attribute, 382
- Instances
 - saving, 50
 - working with, 51
- Integrated systems for generic frameworks, 483
- Integration tests, 485
- IntelliSense, 118, 257–258
- Intentional Software, 14
- Interface classes in UML, 34
- Internal consistency checks, 476–478
- Internal enumerations, 119
- Interpreter pattern, 13
- Interpretive applications, 75–76
- Interpretive architectures, 56
- Interpretive frameworks, 478–484
- InUndoRedoOrRollback method, 406
- Invariants, constraints as, 276
- InvestmentCalculator application, 27–31
- Invoking validation, 288–289
- Is Browsable setting, 120, 190, 192, 405
- Is Custom setting, 223, 227, 271–272, 399
- Is Element Name setting, 98, 439
- Is Moniker Key setting, 261
- Is Property Generator setting, 122
- Is Single Compartment Header Visible setting, 150
- Is UI Read Only setting, 120, 192
- IsConstructorParameter setting, 340
- IsCustom switch, 400
- IsElement Name setting, 120
- IsEmbedding setting, 129
- IServiceProvider mechanism, 428
- IsExpanded setting, 415
- IsStartState setting, 171–172
- Issue class, 327
- Issue State domain model
 - classes in, 92–98, 115–118
 - designer, 60
 - Domain Model Designer for, 88–89
 - generated code for, 109–115
 - notation, 471–474
 - properties in, 95–98, 119–122
 - relationships in, 98–101
 - derivation, 106–108, 126–128
 - embeddings, 101–103
 - inherited, 468
 - link access, 124–126
 - multiplicities in, 100–104
 - reference, 105–106
 - roles in, 122–129
 - snapshots, 460–461
 - issue state, 462–464
 - project and field, 461–462
 - project definitions, 464–468
 - Store for, 89–92, 129–131
- Issue Tracker Starter Kit application
 - domain model for, 48–49
 - evolving, 55–56
 - framework driving in, 51–53
 - generalizing, 42–46
 - refining, 50–51
 - top-down and bottom-up approach, 46–48
 - using, 53–55
- IssueCategory class, 210–211
- IssueProject class, 403
- issueProjectModel directive, 330
- IssueState class, 49–51
 - properties, 91–92, 95–96, 268
 - RequiredFields list, 56
 - serialization class for, 264
- <issueState> elements, 241–242
- <issueStateDiagram> element, 253
- IssueStateModel, 48–50, 270
- IssueStateModel class, 240, 282
- <IssueStateModel> element, 240–241
- IssueStateModelHasComments relationship, 241–242, 262–263
- IssueStateModelHasStates relationship, 48, 241–242
- IssueStateModelsDomainModel class, 115, 130
- IssueStateModelsSerializationHelper class, 265

IssueStateModelsSerializationHelperBase class, 265
 <issueStateMoniker> element, 241
 IssueStateSerializer class, 264, 267–268
 IssueStateTransition class, 124–125, 270
 <issueStateTransition> element, 244
 IssueStateTransition relationship, 105, 241, 243–246
 IsValidIdentifier method, 281
 ITextTemplating interface, 362
 ITextTemplatingEngineHost interface, 355

J

Jacobson, Ivar, 34

K

Kay, Alan, 32
 Keys
 element, 246–248
 package load, 388–390
 Kind setting, 98, 121, 400, 405–406

L

Labels
 editing, 65
 explorer, 183–184
 Language-Oriented Programming, 13
 language parameter, 342
 Language Workbenches, 14
 Languages and language definitions, 2–3
 evolution, 12, 17
 in good DSLs, 492–493
 templates, 57–59
 testing, 489
 LanguageSchema.xsd file, 83
 Large-scale artifact generation, 349–351
 Layers in tool architecture, 79–80
 Layout settings for shapes, 147, 149, 154
 Legacy systems in UML, 38
 Libraries for templatization, 340–341
 License agreements, 77, 377
 <licenseAgreement> element, 382
 Line thickness for shapes and connectors, 411
 Link directives, 207
 adding, 218
 multiple, 219–221
 paths, 221–222
 Link-to-element path segments, 158–159

LinkConnectDirective, 400
 LinkedElementCollection class, 112–113, 123

Links

accessing, 124–126
 creation paths, 205–207
 derived relationships, 108
 in domain models, 462
 embedding, 105, 124, 197, 201–202, 207
 between links, 92
 many-to-many relationships, 109
 references to, 249–251

Load keys, package, 388–390

Load method

DocData, 264–265
 NoLoopClassDocData, 304

Loading

failures, 303–307
 models and diagrams, 238
 LoadModel method, 265–266, 353–354
 LoadModelAndDiagram method, 265
 LocalCommit setting, 413
 localeId attribute, 383

Localization issues in graphical notation, 140–141

LogError method, 293–294

LogErrors method, 355

LogWarning method, 287

Looped inheritance, 299–307

M

manuallyRouted attribute, 254
 Many-to-many relationships, 106, 109
 Maps
 connector, 165–167
 decorator, 170–172
 shape. *See* Shape maps
 Master templates, 350–351
 MDA (Model-Driven Architecture), 13
 Menu commands, 429
 building and running, 435
 command sets for, 431–432
 handlers for
 defining, 432–435
 registering, 440–441
 ids for, 430
 resource indexes for, 430–431
 Merge directives. *See* Element merge directives (EMDs)
 MergeConnect method, 438
 MergeDisconnect method, 212

- MergeDisconnectComment method, 211
 - <mergeModules> element, 384–385
 - MergeRelateComment method, 211
 - Merging and unmerging swimlanes, 162–163
 - Meta-moments, 55
 - Methods in partial classes, 397
 - Microsoft.VisualStudio.Modeling
 - framework, 79
 - Microsoft.VisualStudio.Modeling.Diagrams
 - framework, 80
 - Microsoft.VisualStudio.Modeling.Shell
 - framework, 81
 - Microsoft.VisualStudio.Modeling.TextTemplating
 - framework, 80
 - Microsoft.VisualStudio.Modeling.Validation
 - framework, 80
 - Migration
 - in generated vs. interpretive frameworks, 480
 - serialization, 254–257
 - Minimal language
 - designer for, 62
 - template for, 58–59
 - Minimality in good DSLs, 492
 - Model-Driven Architecture (MDA), 13
 - Model-driven development tools, 13
 - Model elements (MELs). *See* Elements
 - Model setting
 - IssueStateModel, 49
 - StateElement, 111
 - ModelAttribute class, 405–408
 - ModelClass setting, 364
 - ModelElement class, 90–91, 129–130, 413
 - ModelExplorer.cs file, 85
 - ModelHasComments relationships, 207
 - ModelHasProjects relationships, 200–201, 207, 210
 - ModelingTextTransformation class, 347
 - Moniker Attribute Name setting, 261
 - Moniker Element Name setting, 259
 - Moniker Type Name setting, 261
 - MonikerResolver class, 264
 - Monikers, 241, 243, 251
 - for broken diagram files, 256–257
 - calculating and resolving, 270
 - converting, 264
 - in cross-referencing, 245–246
 - load failures with, 255
 - Move Down option, 94
 - Move Up option, 94
 - MoveNestedStates method, 452–453
 - MoveRule class, 414
 - Moving shape children, 451–453
 - .msi files, 77, 387
 - for setup projects, 374
 - for text templates, 391–393
 - with WiX, 371–372
 - MSMs (Windows Installer Merge Modules), 384
 - Multiline text decorators, 173–174
 - Multiple element merge directives, 202
 - Multiple link connect directives, 219–221
 - Multiple link creation paths, 205–207
 - Multiple Projects feature, 459
 - Multiple source and target role directives, 217–219
 - Multiplicities
 - explorer behavior for, 182–183
 - many-to-many relationships, 106
 - names, 113
 - with Omit Element setting, 263
 - in relationship derivation, 126–127
 - relationships among, 474
 - on roles, 100–104, 112–113, 123, 126–127
 - MultiplicityValidation.cs file, 83
- ## N
- name attribute
 - <fileExtension> element, 381
 - <supportingFiles> element, 382
 - Name role, 159
 - Name setting, 98
 - IssueState, 91, 93
 - StartElement, 91
 - StateElement, 96
 - NameAndType setting, 405–406, 408
 - Named Domain Class tool, 93
 - Named Element class, 296
 - NamePropertyHandler class, 296
 - Names
 - elements, 259–261
 - properties, 100, 113
 - in properties window, 190–191
 - qualified, 248–249
 - relationships, 101
 - roles, 113
 - tools, 199
 - Namespace setting, 115
 - Namespaces in folders, 211
 - Nested box notation, 496–497

Nested child shapes, 442–446
 Nested paths notation, 497–498
 <nestedChildShapes> element, 253
 NestedChildShapes relationship, 442

Nesting

embeddings for, 103
 relationship notation for, 469
 .NET event handlers, 419–420
 .NET properties, 193
 .NET remoting, 353

New Project dialog, 57

Next setting, 127

Nodes

appearance, 184–186
 hiding, 186–188
 NodeShape class, 413, 415, 449–450

NoLoopClassDocData class, 304

Notation, 24, 468–469

background pictures in, 176–177

behavior, 22–23

candidate, 495–497

connection points, 177–178

connectors, 164–167, 178–180

decorators, 167–174

diagram definitions, 138–140

editors for, 141–146

familiar, 474–475

in good DSLs, 493–494

issue state, 471–474

localization issues, 140–141

overview, 134–137

project definition, 470–471

regular expressions, 493–494

shapes. *See* Shapes

structure, 21–22

NotifyValueChange method, 121–122,
 407–408

Nullable types, 332

O

Object Constraint Language (OCL), 280–281

Object Modeling Technique, 34

Object-Oriented Software Engineering
 method, 34

ObjectFlow relationship, 219–221

ObjectInState class, 219–220

Observers in validation, 293

Obsolete domain classes, 491

OCL (Object Constraint Language), 280–281

Omit Element setting, 263

OnAbsoluteBoundsChanged method,
 416, 424

OnAssociatedPropertyChanged method,
 408–411

OnBoundsFixup method, 401

OnCopy method, 401

OnDeleted method, 401

OnDeleting method, 401

OnDocumentLoaded method, 402, 418

OnDoubleClick method, 401, 420

One multiplicity

with Omit Element setting, 263

on roles, 102, 104, 123, 126–127

OneMany multiplicity, 104, 112, 126

OnInitialize method, 402

OnMenu handlers, 432–433, 435

OnMenuChangeAssociationSort method,
 432–434

OnMenuCopy method, 438

OnMenuCustomValidate method, 291

OnMenuGenerateClasses method, 360–362

OnMenuPaste method, 439

OnPostLoadModel method, 259

OnPostLoadModelAndDiagram
 method, 259

OnStatus handlers, 432, 435

OnStatusChangeAssociationSort method,
 432–433

OnStatusCopy method, 441

OnStatusPaste method, 441

OnValueChanged method, 401, 403,
 406–408, 411

OnValueChanging method, 296, 401–403, 406

openAfterInstall attribute, 382

Ordina Systems Integrator, 5–8

Output directive, 343

Overrides

custom, 233, 401–402

event, 420

P

Package.cs file, 85, 426

Package load keys (PLKs), 388–390

Packages, 370, 426

for designers, 63–64

registering, 427–428

Parent Element Path setting, 156–157, 443

Parenting shapes, 156

ParentShape property, 423

Parnas, David, 13

- Parsers, 15–16
- Partial classes, 116–117, 397–398
- Partial keyword, 349
- Partitions, 130
- Paste method, 438–440
- Path to display property setting, 161, 170
- Path To Property setting, 186
- Paths
 - domain, 187
 - in element merge directives, 205–207
 - nested, 497–498
 - property, 186
 - role directives, 221–222
 - syntax, 157–160
- Per-class relationships, 410
- Performance
 - in generated vs. interpretive frameworks, 479–480
 - testing, 485
- Phases, 54–55
- Pictures in diagrams, 176–177
- PLKs (package load keys), 388–390
- PopIndent method, 337, 348
- Port tool, 207
- Ports, 153
 - bounds rules on, 423
 - creating, 207
 - decorator positions for, 170
 - mapping, 161–162
- Positioning decorators, 169–170
- Predecessors setting, 49, 127
- Presentation, 133–134
 - explorer. *See* Explorer
 - notation. *See* Notation
 - properties window, 188–194
- PresentationViewsSubject relationship, 408
- Previous setting, 127
- Priority setting, 413
- Private validation methods, 283
- ProcessTemplate method, 355, 362
- Product.ico file, 378
- Product lines, 456
- productCode attribute, 380
- productUrl attribute, 380
- productVersion attribute, 380
- Programmability in generated vs. interpretive frameworks, 481–482
- project attribute
 - <supportingFiles> element, 382
 - <vsItemTemplates> element, 383
- Project class, 209
- Project contents, 60–61, 81–85
- Project definitions, 77–78
 - domain models, 464–468
 - notation, 470–471
 - snapshots, 461–462
- ProjectCategoryConnector class, 179–180
- ProjectDefinition class, 314
- ProjectDefinitionSerializationHelper class, 315
- Propagate flag, 230–231
- PropagateDelete flag, 234
- Propagation
 - change, 408–411
 - delete
 - controlling, 231–232
 - customizing, 232–234
 - rules, 229–230
- Properties
 - calculated, 121, 404–405, 407
 - classes, 116–117
 - domain, 119–122
 - domain models, 49, 95–98, 110–112
 - DomainPropertyInfo, 121–122
 - in DSL evolution, 489
 - forwarding, 189, 192–193
 - hiding, 192
 - meanings, 98
 - names, 100, 113
 - read-only, 192
 - serialization, 242
 - shapes and connectors, 136, 189–191
 - storage, 405–407
 - weakly typed, 280
- Properties window, 63
 - appearance, 188–190
 - categories, names, and descriptions in, 190–191
 - customizing, 72–73, 193
- Property Display Name setting, 185
- Property Displayed path, 185
- Property Getter Access Modifier, 123
- Property handlers for change, 402–404
- Property Name role, 159
- Property paths for nodes, 186
- Property Setter Access Modifier, 123
- Prototypes
 - element tools, 212–216
 - for generic frameworks, 482–483
 - instances, 51

ProvideLoadKey attribute, 389
 ProvideMenuResource attribute, 431
 ProvidePackageLoadKey attribute, 389
 Provider endpoints, 27
 Public enumerations, 119
 PushIndent method, 337, 348

Q

Qualified names, 248–249
 Query View Transformation (QVT)
 initiative, 323

R

Railroad track notation, 495–496
 Range in generated vs. interpretive
 frameworks, 481
 Re-parenting with element merge directives,
 211–212
 Read method, 266
 IssueStateModel, 270
 IssueStateSerializer, 267
 Read-only properties, 192
 ReadChildElements method, 269–270
 ReadElements method, 267, 269
 ReadIssueStateTransitionInstances
 method, 270
 Readme files, 77, 374, 377, 383
 ReadOnlyCollection method, 125
 ReadPropertiesFromAttributes
 method, 267–269
 ReadRootElement method, 266
 Real-world artifact generation, 349–351
 Rectilinear connector routing style, 165
 Redo handler, 403–404
 Redo method, 114
 Redundancy in good DSLs, 492
 Refactoring in DSL evolution, 490–491
 Reference Relationship tool, 99, 102, 107
 Reference relationships
 creating, 99, 107
 in domain models, 49
 overview, 105–106
 References
 Guids as, 246–247
 to links, 249–251
 qualified names as, 248–249
 Refreshing installation files, 387
 regedit.exe command, 370
 Regexp Tree notation, 495

Region issues in notation, 140–141
 Registering
 menu handlers, 440–441
 packages, 427–428
 rules, 130, 417, 448–449
 VSIP partners, 388
 RegistrationAttribute class, 427
 Registry, 370–371
 RegistryRoot attribute, 381
 Regular expressions, 1, 493–494
 Relationships, 24, 49, 98–101, 122–124
 complex, 321–324
 constraints for, 68–70
 derivation, 106–108, 126–128, 245
 in domain models, 48–50
 in DSL evolution, 490
 element merge directives for, 200–201
 embeddings, 49, 101–103, 124
 explorer behavior for, 182
 inherited, 466–468
 link access, 124–126
 mapping connectors to, 166
 multiplicities in. *See* Multiplicities
 notation for, 469
 issue state, 471–474
 project definition, 470
 per-class, 410
 reference, 49, 99, 105–107
 serialization, 242–245
 RelativeChildShapes relationship, 442
 RELAX NG technology, 18
 Remove method, 229
 RemovePassThroughShapes
 method, 301–302
 Representation. *See* Notation
 Representation setting, 261
 Require/Provides pattern, 344
 Required Fields list, 56
 requires parameter, 345
 requiresCPlusPlus attribute, 380
 requiresCSharp attribute, 380
 requiresJSharp attribute, 380
 requiresVisualBasic attribute, 380
 RequiresWatermark method, 402
 ResolvePath method, 356
 Resource indexes for menu commands,
 430–431
 Resources and .resx files, 117
 for image and icons, 152
 for validation messages, 283

- Resources folder, 82
- Resources settings
 - diagrams, 139
 - shapes, 147
- Reverses Direction setting, 225, 227
- Ripple effects in relationships, 321–323
- Role directives
 - connection builders, 217–219
 - paths, 221–222
- Role Element Name setting, 262
- RolePlayerChangeRule class, 414
- RolePlayerPositionChangedRule class, 414
- Roleplayers, 100
- Roles, 54–55, 100–101
 - multiplicities on, 100–104, 112–113, 123, 126–127
 - names, 113
 - relationships, 49
- Rollback method, 114
- Root domain classes, 92
- Rotate setting, 175
- Round-tripping for artifacts, 321–324
- Routing style of connectors, 165, 178–180
- RuleOn attribute, 130, 303, 413–414
- Rules
 - bodies, 415
 - bounds, 420–424, 448–449
 - calculated and custom storage properties, 416
 - change, 412–413, 449–451
 - connector maps, 167
 - delete propagation, 229–230
 - hard constraints, 296–298
 - model changes, 322
 - in partitions, 130
 - registering, 130, 417, 448–449
 - RuleOn attribute, 130, 303, 413–414
 - shape containers, 446–453
 - shape maps, 164
 - stores, 121
 - testing, 489
 - transactions, 416–417
 - types, 414–415
- Rumbaugh, James, 34
- Run Custom Tool option, 332, 488
- Running menu commands, 435
- Runtime
 - access, 118
 - CLR, 37, 120
 - DSL, 54

S

- Save method
 - DocData, 264–265
 - for templates, 326, 332–337
- SaveModel method, 265
- SaveModelAndDiagram method, 265
- Saving
 - instances, 50
 - models and diagrams, 238
 - serialization for. *See* Serialization
- Schematron.NET, 18
- Scope in good DSLs, 491
- SDM (System Definition Model), 20–21, 29
- Sealed domain classes, 95
- Segments, path, 158–159
- Semicolon-delimited lists, 345
- Separation of domain models and presentation, 49–50
- Sequence diagrams, 34, 37, 39
- Serialization, 25, 73–74
 - cross-referencing, 245–251
 - customization, 258–264
 - customized code, 271–272
 - diagrams, 251–254
 - element ids, 129
 - elements and properties, 242
 - generated code, 264–270
 - implementing, 264
 - overview, 237
 - relationships, 243–245
 - transactions, 265
 - versioning and migration, 254–257
 - XML file format for. *See* XML (Extensible Markup Language)
 - XML Schema file for, 257–258
- SerializationHelper.cs file, 83, 264–265, 315
- Serialize Id setting, 260–261
- Serializer.cs file, 83
- Serializers, 60
- ServerName setting, 294
- Service DSL, 7
- SetElementSerializer class, 267
- SetNameAndTypeValue method, 406–407
- Setter Access Modifier setting, 120
- settings.ini file, 374, 377
- SetUniqueName method, 120
- setup.exe file, 372, 374–375
- setup.ini file, 376

- Setup projects
 - contents, 376–377
 - creating, 373–376
 - customizing, 377–378
- SetValue method, 122
- SetXXXValue method, 121
- Shape containers, 442
 - child shapes, 442–446
 - rules for, 446–453
- Shape maps, 50, 154–155
 - advanced, 163–164
 - for all shapes, 155–157
 - for compartments, 160–161
 - creating, 155
 - for elements, 136
 - for ports, 161–162
 - for swimlanes, 162–163
- ShapeElement class, 413
- ShapeMap switch, 400
- Shapes, 146
 - categories, 137, 147
 - child, 442
 - moving, 451–453
 - nested, 443–446
 - compartment, 149–151
 - for decorators, 173
 - in domain models, 49–50
 - domain models without, 108–109
 - geometry, 148–149
 - image, 151–152
 - inheritance, 154
 - parenting, 156
 - per-class relationships involving, 410
 - ports, 153
 - propagating change to, 408–411
 - properties for, 136
 - style properties on, 189–191
 - swimlanes, 153–154
 - variable, 174–176
 - Z Order, 453
- Shapes.cs file, 83
- Shell architecture, 426–429
- shortcutIconPath attribute, 382
- ShouldVisitRolePlayer method, 232–233
- Show as Class option, 107
- SimpleMonikerResolver class, 246
- Single arrowheads, 110–112
- Size of shapes, 136
 - compartment, 150
 - image, 152
 - swimlanes, 154
- SMART-Microsoft Software Factory, 6
- SML (System Modeling Language), 21
- Snapshots, 460–461
 - issue state, 462–464
 - project and field, 461–462
 - project definitions, 464–468
 - for test cases, 487
- Soft constraints, 226, 280–281
 - vs. hard, 277–280
 - with hard, 299–307
- Software defined circuitry, 8–9
- Software development customization, 9
- Software factories, 14–15
- SortDomainPropertyId, 410
- SortPropertyHandler class, 403
- Source ends for connectors, 165
- Source role directives, 217–219, 227
- Sources of domain relationships, 98
- Split Tree option, 94
- Standard control blocks, 346–347
- Start Element setting, 134
- StartElement class, 49, 51, 91–93
- <startElement> element, 241
- StartPoints in state charts, 446–447
- StartTransition relationship, 107
- State charts, 446–447
- State setting, 49
- StateChangeHandler method, 418
- StateChartsDomainModel class, 417
- StateElement class
 - in IssueStateModel, 48–49
 - Model property, 111
 - Name property, 96
 - relationships with, 99–100
- States
 - issue state notation, 471–474
 - state charts, 446–447
 - <states> element, 241–242
- StateShape class, 419–420, 449
- StateShape_DoubleClick method, 419–420
- StateShapeBoundsRule class, 448
- Stdidcmd.h file, 435
- StockPriceApplication application, 27–31
- StockPrices.asmx file, 29
- StockPrices.cs file, 28, 30
- StockPrices service, 27–31
- Store, 89–92
 - DocData for, 428
 - DomainModelInfo for, 130–131

- element lookup in, 129
- events, 121, 417–419
- partitions in, 130
- types in, 481
- unresolved references in, 245
- Store setting, 347
- Straight connector routing style, 165
- StringBuilder setting, 347
- Strings
 - for paths, 159–160
 - for properties, 49, 96, 119–120
- Strings.wxl file, 376, 378
- Structures in diagrams, 21–22
- Style
 - connector routing, 165, 178–180
 - shapes and connectors, 136, 189–191
- Sub-parameters in custom directives, 345
- <subjects> element, 241
- SubModel setting, 346
- <successors> element, 241
- Successors setting, 49, 127
- SuperClass setting, 299
- <supportingAssemblies> element, 381
- <supportingFiles> element, 382–383
- Swimlanes, 153–154
 - decorator positions for, 170
 - mapping, 162–163
- Switches, customization, 399–400
- SyncBarBoundsRule class, 422
- SyncBarShape class, 178, 421–422
- Synchronization bar, 420–421
- Synchronization schemes, 323
- Syntactic space, 492
- Syntax
 - path, 157–160
 - text templates, 341–349
- System Definition Model (SDM), 20–21, 29
- System Modeling Language (SML), 21

T

- Target ends for connectors, 165
- Target role directives, 217–219, 227
- targetDirectories attribute, 383
- Targets of domain relationships, 98
- Task Flows template, 58–59, 151
- TaskValidationMessage class, 294
- temp directory, 342
- template directive, 329–330, 341–342
- templatePath attribute, 383
- Templates and templization process, 57–59, 325–328
 - for artifacts, 319–320
 - custom hosting, 354–357
 - custom orchestration, 359–366
 - data structures in, 350
 - directive processor for, 357–359
 - editor, 320
 - first cut, 328–337
 - generic-specific model data in, 338–340
 - libraries for, 340–341
 - testing, 488
 - text. *See* Text templates
- Test languages, 134
- Test models, 486
- Testing, 484–489
 - generated code, 488–489
 - generator templates, 488
 - language definitions, 489
 - rules, 489
 - validation constraints, 486–487
- TestValidInheritance method, 298, 302
- Text Color setting, 149
- Text decorators, 168–169
 - filtering, 171–172
 - multiline, 173–174
- Text templates, 57, 320
 - architecture, 352–354
 - for code generation, 390–395
 - control blocks for, 346–349
 - coverage analysis, 488
 - from Debugging project, 390–392
 - directives in, 341–346
 - include file, 392–393
 - syntax of, 341–349
 - in VS Item Template, 393–395
- <textTemplates> elements, 385
- TextTemplatingFileGenerator, 394
- TextTransform.exe script, 354–355
- TextTransformation class, 347
- Textual DSLs, 15–20
- Textual languages, 2–3, 11
- Title setting, 183
- Toolbox, 62
- ToolboxHelper.cs file, 83
- Tools
 - architecture, 78
 - Dsl project contents, 81–83
 - DslPackage project contents, 83–85
 - framework assemblies, 79–81

Tools (*Continued*)

- generated code, 78–79
- layers, 79–80
- for elements, 198–200, 212–216
- integrating, 26
- Tooltips, 199
- TooltipType switch, 400
- ToolWindow class, 428
- Top-down approach
 - in incremental development, 46–48
 - in variability identification, 457
- Topic areas, 462
- TopLevelCommit, 413
- ToString method, 349
- TransactionBeginningRule class, 415
- TransactionCommittingRule class, 415
- TransactionRollingBackRule class, 415
- Transactions, 114
 - context, 416–417
 - rules in, 416–417
 - serializing, 265
- Transform All Templates option, 66
- Transformation class, 353, 357–358
- Transforms, 311–314
- TransformText method, 353–354, 358
- Transistor class, 175
- Transistor tool, 213
- TransistorShape class, 176
- TransistorShape tool, 174–176
- Transition relationships, 49–50, 126, 245
- TransitionConnector, 50, 137
- Translation tools, 491
- Trask, Bruce, 9, 11
- .tt extension, 51, 61, 394–395
- Type-based constraints, 279
- Type Name setting, 261
- TypePropertyHandler class, 407–408
- Types
 - in generated vs. interpretive frameworks, 481–482
 - properties, 96, 119–120

U

- UML (Unified Modeling Language), 34–40
- Undo handler, 403–404
- Undo method, 114
- Undoable changes, 424
- Unified Modeling Language (UML), 34–40
- Unit tests, 484
- Unmapped swimlanes, 163

Unmerging

- element merges, 212
- swimlanes, 162–163
- Update plans in DSL evolution, 491
- UpdateButton_Click method, 145–146
- UpdateParent method, 450–451
- Upfront investment in generated vs. interpretive frameworks, 480
- upgradeCode attribute, 380
- Use cases, 4–6, 34–40
- Use custom element collection option, 161
- Use Full Form setting, 262
- UseCustomElement switch, 400
- User Control files, 435–436
- User Control template option, 435
- User Profile Model, 4
- Users, 54–55
- Uses custom accept flag, 208–210
- Uses custom connect option, 223, 227
- Uses custom merge flag, 210–211
- Uses Custom setting, 289
- UsesCustomAccept switch, 400
- UsesCustomMerge switch, 400
- using statements, 114

V

- Validate All option, 102
- Validate method, 289–290
- ValidateCustom method, 289–291
- ValidateDbConnection method, 294
- ValidateNonCyclical method, 305
- ValidateStateNamesUnique method, 282
- ValidateStatesReachable method, 285–287
- Validation, 51, 70. See also Constraints
 - connection builders, 222
 - custom categories, 289–292
 - defining, 475–478
 - domain models, 89
 - in DSL evolution, 491
 - enabling, 284–287
 - external data, 294–295, 478
 - inheriting behavior, 292
 - internal data, 476–478
 - invoking, 288–289
 - methods, 281–284
 - multiplicity rules, 127
 - output, 292–293
 - outside IDE, 293–294
 - resource files for, 283
 - testing, 486–487

Validation node, 289
 ValidationCategory enumeration, 287–288
 ValidationContext class, 283, 287, 294
 ValidationControllor class, 294
 ValidationMessage class, 292–294
 ValidationMethod attribute, 285, 292
 ValidationObserver class, 292–294
 ValidationState attribute, 284–285, 292
 ValidationSubjects setting, 287
 ValidatonController class, 288–291
 Value property handlers, 295
 ValueChanged method, 406
 ValueChanging method, 406
 Variability
 in design, 456–457
 in domain models, 462
 in incremental development, 42–46
 Variable image shapes, 174–176
 Versioning, 254–257
 Vertical DSLs, 8
 Vertical swimlanes, 153–154
 View fixup, 50
 ViewForm class, 145–146
 Visibility
 decorators, 170–172
 DSL Details window, 155
 shapes and connectors, 411
 VisitorFilterResult method, 233–234
 Visual Basic in templates, 331
 Visual Studio DSL development, 27–31, 57
 adding to DSLs, 67–68
 authoring solutions, 57–60
 code generation, 66–67
 constraints, 68–70
 custom code, 73
 defining DSLs, 64–65
 deployment, 76–77
 driving applications from DSLs, 74–76
 explorer window, 71–72
 file extensions, 63–64
 properties window, 72–73
 serialization format, 73–74
 trying out DSLs, 61–63
 Visual Studio Packages, 370
 Vlissides, John, 398
 VS Item Template, 393–395
 VSIP partners, 388
 <vsItemTemplates> element, 383–384

VSPackage.resx file, 184, 389–390
 <vsProjectTemplates> element, 384
 Vsshids.h file, 435
 VsValidationControllor class, 292, 294

W

Walkthroughs, 357
 Warning method, 347
 WatermarkText method, 402
 Weakly typed properties, 280
 Web Scenario DSL, 6
 Web Service Definition Language (WSDL), 38
 Width
 image shapes, 152
 swimlanes, 154
 Window icons, 183–184
 Windows Forms controls, 142–143
 Windows installer, 371–372
 Windows Installer Merge Modules (MSMs), 384
 Windows Installer XML (WiX), 371–373
 Windows Presentation Foundation (WPF), 4
 Windows registry, 370–371
 WinformDSL1DocView class, 436
 WiX (Windows Installer XML), 371–373
 WPF (Windows Presentation Foundation), 4
 Write method
 IssueStateModel, 270
 for text templates, 347
 WriteLine method, 347
 WSDL (Web Service Definition Language), 38

X

XHasY relationship, 101
 XML (Extensible Markup Language), 25, 73–74, 239–241
 cross-referencing, 245–251
 customization, 258–264
 customized code, 271–272
 diagrams, 251–254
 element names, 259–261
 elements and properties, 242
 generated code, 264–270
 relationships, 243–245
 textual DSLs, 17–20
 versioning and migration, 254–257

Xml Name setting, 262
XML Schema, 18, 257–258, 272–273
XmlReader approach, 266
XReferencesY relationship, 101
XSLT
 for model file migration, 314
 stylesheets, 311–314

Z

Z Order, shapes, 453
ZeroMany multiplicity, 101, 104, 112–113, 126
ZeroOne multiplicity, 102, 104, 123,
 126–127, 263
Zones in diagrams, 20
ZOrder setting, 453