



HIBERNATE

Quickly

Patrick Peak
Nick Heudecker

 MANNING

Contents

preface xv
acknowledgments xvii
about this book xviii
about the cover illustration xxiv

1 Why Hibernate? 1

- 1.1 *Understanding object persistence* 3
 - Identity 4 ◦ Inheritance 5 ◦ Associations 5
 - Object/relational mapping 6
- 1.2 *Using direct JDBC* 9
 - Example application 10 ◦ Retrieving object graphs using JDBC 11 ◦ Persisting object graphs to a relational model 15 ◦ Deleting object graphs 18
 - Querying object graphs 19
- 1.3 *Persistence with Hibernate* 20
 - Simplicity and flexibility 20 ◦ Completeness 22
 - Performance 23
- 1.4 *Summary* 25

2 Installing and building projects with Ant 26

- 2.1 *Getting a Hibernate distribution* 28
 - Installing Ant 30 ◦ Getting Ant 30 ◦ Extracting and installing Ant 30

- 2.2 *Setting up a database* 31
 - Getting MySQL 32 ◦ Testing MySQL 32
 - MySQL drivers 34
 - 2.3 *Setting up a project* 34
 - Defining directories 35 ◦ Ant 101 36
 - Running Ant 39
 - 2.4 *Habits of highly effective build files* 41
 - Connecting Hibernate 42 ◦ Reusable build files 46 ◦ Expanding your horizons 48
 - 2.5 *Summary* 49
- 3 Hibernate basics 50**
- 3.1 *Configuring Hibernate* 51
 - Basic configuration 53
 - 3.2 *Creating mapping definitions* 56
 - IDs and generators 58 ◦ Properties 59
 - Many-to-one element 60 ◦ Proxies 62
 - Collections 63 ◦ Cascades 64
 - Fetching associated objects 66
 - 3.3 *Building the SessionFactory* 66
 - Configuring the SessionFactory 66
 - 3.4 *Persisting objects* 68
 - 3.5 *Retrieving objects* 70
 - 3.6 *The Session cache* 72
 - 3.7 *Advanced configuration* 74
 - Connection pools 74 ◦ Transactions 76
 - Cache providers 79
 - 3.8 *Inheritance* 83
 - Table per class hierarchy 83
 - Table per subclass 85
 - 3.9 *Summary* 86

4 Associations and components 88

4.1 Associations 89

- Many-to-one relationships, in depth 90
- The central configuration file 95
- Defining sample data 96

4.2 Building tables with Ant and SchemaExport 99

- Logging with log4j and Commons Logging 102
- Running SchemaExport 104
 - Loading the Events 106
 - Refactoring 108
 - Finding Events 113
 - Cascades 115

4.3 Components 116

- What's in a component? 117
 - Mapping a component 119
 - Why use a component? 121

4.4 Summary 122

5 Collections and custom types 123

5.1 Persisting collections and arrays 125

- Using interfaces 126
 - Mapping persistent collections 128
 - Collection types 131
- Lazy collections 138
 - Sorted collections 139
- Bidirectional associations 142
 - Cascading collections 145

5.2 Implementing custom types 147

- UserTypes 148
 - Implementing CompositeUserTypes 154

5.3 Summary 159

6 Querying persistent objects 161

6.1 Using HQL 162

- `session.find(...)` 163
 - The Query interface 164
- Outer joins and HQL 168
 - Show SQL 169
- Query substitutions 169
 - Query parser 170

- 6.2 *Querying objects with HQL* 171
 - The FROM clause 171 ◦ Joins 172
 - Selects 174 ◦ Using functions 176
 - HQL properties 178 ◦ Using expressions 179
- 6.3 *Criteria queries* 183
- 6.4 *Stored procedures* 185
- 6.5 *HibernateIDE* 186
- 6.6 *Summary* 187

7 Organizing with Spring and data access objects 189

- 7.1 *The ubiquitous DAO pattern* 190
 - Keeping the HQL together 191
- 7.2 *Analyzing the DAO* 196
 - Boilerplate code 196 ◦ Potential duplication 196 ◦ Detached objects only 197
- 7.3 *The Layer Supertype pattern* 198
 - Creating an AbstractDao 199
- 7.4 *The Spring Framework* 202
 - What's in a template? 204
 - Beans and their factories 208
- 7.5 *Summary* 215

8 Web frameworks: WebWork, Struts, and Tapestry 217

- 8.1 *Defining the application* 219
- 8.2 *A quick overview of MVC* 219
 - Service Layer pattern 222
- 8.3 *Decoupling Hibernate from the web layer* 227
 - Working with detached objects 227 ◦ Session scope 229 ◦ Accessing the Session from the Controller 230 ◦ Accessing the Session from the Service layer 235

- 8.4 *WebWork* 238
 - WebWork fundamentals 238
 - Creating controllers 239
- 8.5 *Struts* 253
 - Struts fundamentals 254
 - Building Struts Actions 256
- 8.6 *Tapestry* 261
 - Getting started 261 ◦ Tapestry fundamentals 261 ◦ HTML views 262
 - Page controller 264 ◦ Page specification 267
 - web.xml 269
- 8.7 *Hibernate in the view layer* 270
- 8.8 *Summary* 272

9 Hibernating with XDoclet 274

- 9.1 *Essential XDoclet* 276
 - JavaDoc basics 276 ◦ XDoclet: Building your own tags 277 ◦ Installing XDoclet 279
 - Configuring Ant 280
- 9.2 *Making single objects persistent* 282
 - The @hibernate.class tag 283 ◦ The @hibernate.id tag 284 ◦ The @hibernate.property tag 287
 - The @hibernate.column tag 289
- 9.3 *Basic relationships* 292
 - The @hibernate.many-to-one tag 292
 - The @hibernate.component tag 295
- 9.4 *Building collections* 300
 - One-to-many: a kicking set of Speakers 301
 - The @hibernate.set tag 303
 - The @hibernate.collection-key 304
 - The @hibernate.collection-one-to-many tag 305
- 9.5 *Going where no XDoclet has gone before* 306
 - Merge points 306 ◦ Property substitution 308
- 9.6 *Generating the hibernate.cfg.xml file* 310
- 9.7 *Summary* 311

10	Unit testing with JUnit and DBUnit	313
10.1	<i>Introduction to unit testing</i>	314
	Automate those tests	315
	Assertions	316
	Expect failures	317
10.2	<i>JUnit</i>	318
	Test-infecting your build file	318
	Polishing off the build file	321
10.3	<i>Testing the persistence layer</i>	324
	What do we want to test?	324
	Testing basic persistence	325
	Testing queries	329
	General database testing tips	331
10.4	<i>Testing with DBUnit</i>	336
	Loading test data	336
	ProjectDatabaseTestCase	340
10.5	<i>Summary</i>	345
11	What's new in Hibernate 3	346
11.1	<i>Filters</i>	347
11.2	<i>Mapping improvements</i>	348
	Multiple table mapping	348
	Discriminator formulas	349
	Union subclasses	351
	Property references	352
11.3	<i>Dynamic classes</i>	352
11.4	<i>Annotations</i>	354
11.5	<i>Stored procedures and SQL</i>	357
11.6	<i>Persistence events</i>	359
11.7	<i>Lazy properties</i>	361
11.8	<i>Summary</i>	363
Appendix	The complete Hibernate mapping catalog	364
	Index	411