

Hibernate

*A Developer's
Notebook™*

Hibernate 程序高手秘笈
(影印版)

James Elliott 著

- Easy persistence
- Query-free code
- Automate plumbing
- Database neutral
- Flexible & extensible

O'REILLY®

東南大學出版社

Hibernate

程序高手秘笈

(影印版)

江苏工业学院图书馆
藏书章

James Elliott

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

图书在版编目 (CIP) 数据

Hibernate 程序高手秘笈 / (美) 埃利奥特 (Elliott, J.)
著. —影印本. —南京: 东南大学出版社, 2005.10

书名原文: Hibernate: A Developer's Notebook

ISBN 7-5641-0166-0

I. H... II. 埃... III. JAVA 语言—程序设计—英文 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 101833 号

江苏省版权局著作权合同登记

图字: 10-2005-199 号

©2004 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2005. Authorized reprint of the original English edition, 2004 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2004。

英文影印版由东南大学出版社出版 2005。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

书 名 / Hibernate 程序高手秘笈 (影印版)

书 号 / ISBN 7-5641-0166-0

责任编辑 / 张烨

封面设计 / Edie Freedman, 张健

出版发行 / 东南大学出版社 (press.seu.edu.cn)

地 址 / 南京四牌楼 2 号 (邮政编码 210096)

印 刷 / 扬中市印刷有限公司

开 本 / 787 毫米 × 980 毫米 16 开本 12 印张

版 次 / 2005 年 11 月第 1 版 2005 年 11 月第 1 次印刷

印 数 / 0001-2000 册

定 价 / 32.00 元

O'Reilly Media, Inc.介绍

O'Reilly Media, Inc.是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的 50 本书之一）到 GNN（最早的 Internet 门户和商业网站），再到 WebSite（第一个桌面 PC 的 Web 服务器软件），O'Reilly Media, Inc.一直处于 Internet 发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc.是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc.具有深厚的计算机专业背景，这使得 O'Reilly Media, Inc.形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc.所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc.还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc.依靠他们及时地推出图书。因为 O'Reilly Media, Inc.紧密地与计算机业界联系着，所以 O'Reilly Media, Inc.知道市场上真正需要什么图书。

出版说明

随着计算机技术的成熟和广泛应用,人类正在步入一个技术迅猛发展的新时期。计算机技术的发展给人们的工业生产、商业活动和日常生活都带来了巨大的影响。然而,计算机领域的技术更新速度之快也是众所周知的,为了帮助国内技术人员在第一时间了解国外最新的技术,东南大学出版社和美国 O'Reilly Meida, Inc.达成协议,将陆续引进该公司的代表前沿技术或者在某专项领域享有盛名的著作,以影印版或者简体中文版的形式呈献给读者。其中,影印版书籍力求与国外图书“同步”出版,并且“原汁原味”展现给读者。

我们真诚地希望,所引进的书籍能对国内相关行业的技术人员、科研机构的研究人员和高校师生的学习和工作有所帮助,对国内计算机技术的发展有所促进。也衷心期望读者提出宝贵的意见和建议。

已出版影印图书如下:

- 《核心 Java》(影印版)
- 《Jakarta Commons 经典实例》(影印版)
- 《Weblogic 权威指南》(影印版)
- 《Java 网络编程 第三版》(影印版)
- 《Linux 设备驱动程序 第三版》(影印版)
- 《LPI Linux 认证权威指南》(影印版)
- 《GNU Make 项目管理 第三版》(影印版)
- 《游戏开发中的人工智能》(影印版)
- 《学习 Python 第二版》(影印版)
- 《精通正则表达式 第二版》(影印版)
- 《深入浅出设计模式》(影印版)
- 《深入浅出 Java 第二版》(影印版)
- 《轻快的好 Java》(影印版)
- 《Hibernate 程序高手秘笈》(影印版)

Preface

Hibernate is a lightweight object/relational mapping service for Java. What does that mean? It's a way to easily and efficiently work with information from a relational database in the form of natural Java objects. But that description doesn't come close to conveying how useful and exciting the technology is. I'm not the only person who thinks so: Hibernate 2.1 just won *Software Development* magazine's 14th annual Jolt Award in the "Libraries, Frameworks, and Components" category.

So, what's great about Hibernate? Every nontrivial application (and even many trivial ones) needs to store and use information, and these days this usually involves a relational database. Databases are a very different world than Java objects, and they often involve people with different skills and specializations. Bridging between the two worlds has been important for a while, but it used to be quite complex and tedious.

Most people start out struggling to write a few SQL queries, embedding these awkwardly as strings within Java code, and working with JDBC to run them and process the results. JDBC has evolved into a rich and flexible database communication library, which now provides ways to simplify and improve on this approach, but there is still a fair degree of tedium involved. People who work with data a great deal need more power, some way of moving the queries out of the code, and making them act more like well-behaved components in an object-oriented world.

Such capabilities have been part of my own (even more) lightweight object/relational layer for years. It began with a Java database connection and query pooling system written by my colleague Eric Knapp for the Lands' End e-commerce site. Our pooler introduced the idea of external SQL templates that could be accessed by name and efficiently combined with runtime data to generate the actual database queries. Only

later did it grow to include the ability to bind these templates directly to Java objects, by adding simple mapping directives to the templates.

Although far less powerful than a system like Hibernate, this approach proved valuable in many projects of different sizes and in widely differing environments. I've continued to use it to this day, most recently in building IP telephony applications for Cisco's CallManager platform. But I'm going to be using Hibernate instead from now on. Once you work through this book, you'll understand why, and will probably make the same decision yourself. Hibernate does a tremendous amount for you, and does it so easily that you can almost forget you're working with a database. Your objects are simply there when you need them. This is how technology should work.

You may wonder how Hibernate relates to Enterprise JavaBeans™. Is it a competing solution? When would you use one over the other? In fact, you can use both. Most applications have no need for the complexity of EJBs, and they can simply use Hibernate directly to interact with a database. On the other hand, EJBs are indispensable for very complex three-tier application environments. In such cases, Hibernate may be used by an EJB Session bean to persist data, or it might be used to persist BMP entity beans.

How to Use This Book

The Developer's Notebook™ series is a new approach to helping readers rapidly come up to speed with useful new technologies. This book is not intended to be a comprehensive reference manual for Hibernate. Instead, it reflects my own exploration of the system, from initial download and configuration through a series of projects that demonstrate how to accomplish a variety of practical goals.

By reading and following along with these examples, you'll be able to get your own Hibernate environment set up quickly and start using it for realistic tasks right away. It's as if you can "walk with me" through terrain I've mapped out, while I point out useful landmarks and tricky pitfalls along the way.

Although I certainly include some background materials and explanations of how Hibernate works and why, this is always in the service of a focused task. Sometimes I'll refer you to the reference documentation or other online resources if you'd like more depth about one of the underlying concepts or details about a related but different way to use Hibernate.

Once you're past the first few chapters, you don't need to read the rest in order; you can jump to topics that are particularly interesting or relevant to you. The examples do build on each other, but you can download the finished source code from the book's web site (you may want to start with the previous chapter's files and follow along making changes yourself to implement the examples you're reading). You can always jump back to the earlier examples if they turn out to be interesting because of how they relate to what you've just learned.

Font Conventions

This book follows certain conventions for font usage. Understanding these conventions up-front makes it easier to use this book.

Italic

Used for filenames, file extensions, URLs, application names, emphasis, and new terms when they are first introduced.

Constant width

Used for Java class names, methods, variables, properties, data types, database elements, and snippets of code that appear in text.

Constant width bold

Used for commands you enter at the command line and to highlight new code inserted in a running example.

Constant width italic

Used to annotate output.

On the Web Site

The web site for this book, <http://www.oreilly.com/catalog/0596006969>, offers some important materials you'll want to know about. All the examples for this book can be found there, organized by chapter.

The examples are available as a ZIP archive and a compressed TAR archive.

In many cases, the same files are used throughout a series of chapters, and they evolve to include new features and capabilities from example to example. Each chapter folder in the downloadable archive contains a snapshot of the state of the example system, reflecting all the changes and new content introduced in that chapter.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly & Associates, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
(800) 998-9938 (in the United States or Canada)
(707) 829-0515 (international or local)
(707) 829-0104 (fax)

O'Reilly's web page for this book, where we list errata, examples, or any additional information. You can access this page at:

<http://www.oreilly.com/catalog/hibernate/>

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our web site at:

<http://www.oreilly.com/>

Acknowledgments

Any list of thanks has to start with my parents for fostering my interest in computing even when we were living in countries that made that a major challenge, and with my partner Joe for putting up with it today when it has flowered into a major obsession. I'd also like to acknowledge my employer, Berbee, for giving me an opportunity to delve deeply into Java and build skills as an architect of reusable APIs; for letting me stay clear of the proprietary, platform-specific tar pit that is engulfing so much of the programming world; for surrounding me with such incredible colleagues; and for being supportive when I wanted to leverage these experiences in writing this book.

Marc Loy got me connected with the wonderful folks at O'Reilly by inviting me to help with the second edition of *Java Swing*, and Mike Loukides has been patiently working on me ever since—encouraging me to write a book of my own. In Hibernate he found the perfect topic to get me started. Deb Cameron, our revisions editor for the *Swing* effort, has played a big role in turning my tentative authorial ambitions into a rewarding reality. I'm also grateful she was willing to "loan me out" from

helping with the third edition of *Learning Emacs* to take on the Hibernate project.

I'm particularly indebted to my technical reviewers: Adrian Kellor and Curt Pederson. They looked at some very early drafts and helped set my tone and direction, as well as reinforcing my enthusiasm about the value of this project. As the book came together, Bruce Tate provided an important sanity check from someone actively using and teaching Hibernate, and he offered some great advice and even more encouragement. Eric Knapp reviewed a large portion with an eye toward using the book in an instructional setting at a technical college, and reminded me to keep my feet on the ground. Tim Cartwright jumped in at the end, working with a nearly complete draft in an effort to understand Hibernate as a potential platform for future work, and providing a great deal of useful feedback about the content and presentation.

I'd also like to thank the many members of O'Reilly's production department who put in lots of work under an unusually tight schedule.

Contents

Preface	ix
Chapter 1. Installation and Setup	1
Getting an Ant Distribution	1
Getting the HSQLDB Database Engine	3
Getting Hibernate	4
Setting Up a Project Hierarchy	5
Chapter 2. Introduction to Mapping	13
Writing a Mapping Document	14
Generating Some Class	17
Cooking Up a Schema	24
Connecting Hibernate to MySQL	32
Chapter 3. Harnessing Hibernate	38
Creating Persistent Objects	38
Finding Persistent Objects	47
Better Ways to Build Queries	52
Chapter 4. Collections and Associations	55
Mapping Collections	55
Persisting Collections	63
Retrieving Collections	69
Using Bidirectional Associations	71
Working with Simple Collections	76

Chapter 5. Richer Associations.	78
Using Lazy Associations	78
Ordered Collections	80
Augmenting Associations in Collections	82
Lifecycle Associations	94
Reflexive Associations	97
Chapter 6. Persistent Enumerated Types	100
Defining a Persistent Enumerated Type	100
Working with Persistent Enumerations	106
Chapter 7. Custom Value Types	111
Defining a User Type	112
Using a Custom Type Mapping	116
Building a Composite User Type	118
Chapter 8. Criteria Queries	131
Using Simple Criteria	132
Compounding Criteria	135
Applying Criteria to Associations	137
Querying by Example	139
Chapter 9. A Look at HQL	142
Writing HQL Queries	142
Selecting Properties and Pieces	147
Sorting	151
Working with Aggregate Values	152
Writing Native SQL Queries	155
Appendix A. Hibernate Types.	159
Appendix B. Standard Criteria	165
Appendix C. Hibernate SQL Dialects	168
Index	171

Installation and Setup

It continues to amaze me how many great, free, open source Java™ tools are out there. When I needed a lightweight object/relational mapping service for a JSP e-commerce project several years ago, I had to build my own. It evolved over the years, developed some cool and unique features, and we've used it in a wide variety of different contexts. But now that I've discovered Hibernate, I expect that I'll be using it on my next project instead of my own familiar system (toward which I'll cheerfully admit bias). That should tell you how compelling it is!

Since you're looking at this book, you're interested in a powerful and convenient way to bridge between the worlds of Java objects and relational databases. Hibernate fills that role very nicely, without being so big and complicated that learning it becomes a daunting challenge in itself. To demonstrate that, this chapter guides you to the point where you can play with Hibernate and see for yourself why it's so exciting.

In this chapter:

- *Getting an Ant Distribution*
- *Getting the HSQLDB Database Engine*
- *Getting Hibernate*
- *Setting Up a Project Hierarchy*

Getting an Ant Distribution

If you're not already using Ant to manage the building, testing, running, and packaging of your Java projects, now is the time to start. The examples in this book are Ant driven, so you'll need a working Ant installation to follow along and experiment with variations on your own system, which is the best way to learn.

First of all, get an Ant binary and install it.

Why do I care?

We chose to structure our examples around Ant for several reasons. It's convenient and powerful, it's increasingly (almost universally) the

*I used to wonder
why people
bothered with
Ant when they
could use Make.
Now that I've
seen how well it
manages Java
builds, I feel lost
without it.*

standard build tool for Java-based development, it's free and it's cross-platform. Many of the example and helper scripts in the current Hibernate distribution are Windows batch files, which don't do any good for people like me who live in a Unix world. Using Ant means our examples can work equally well anywhere there's a Java environment, which means we don't have to frustrate or annoy any readers of this book. Happily, it also means we can do many more cool things with less effort—especially since several Hibernate tools have explicit Ant support, which we'll show how to leverage.

To take advantage of all this, you need to have Ant installed and working on your system.

How do I do that?

Download a binary release of Ant from <http://ant.apache.org/bindownload.cgi>. Scroll down to find the current release of Ant, and download the archive in a format that's convenient for you to work with. Pick an appropriate place for it to live, and expand the archive there. The directory into which you've expanded the archive is referred to as ANT_HOME. Let's say you've expanded the archive into the directory `/usr/local/apache-ant-1.5.1`; you may want to create a symbolic link to make it easier to work with, and to avoid the need to change any environment configuration when you upgrade to a newer version:

```
/usr/local $ ln -s apache-ant-1.5.1 ant
```

Once Ant is situated, you need to do a couple of things to make it work right. You need to add its `bin` directory in the distribution (in our example, `/usr/local/ant/bin`) to your command path. You also need to set the environment variable ANT_HOME to the top-level directory you installed (in this example, `/usr/local/ant`). Details about how to perform these steps under different operating systems can be found in the Ant manual, <http://ant.apache.org/manual/>, if you need them.

Of course, we're also assuming you've got a Java SDK. Because some of Hibernate's features are available only in Java 1.4, you'd be best off upgrading to the latest 1.4 SDK. It's also possible to use most of Hibernate with Java 1.3, but you may have to rebuild the Hibernate JAR file using your 1.3 compiler. Our examples are written assuming you've got Java 1.4, and they will need tweaking if you don't.

Once you've got this set up, you should be able to fire up Ant for a test run and verify that everything's right:

```
~ $ ant -version  
Apache Ant version 1.5.1 compiled on February 7 2003
```

What just happened?

Well, not much just yet, but you're now in a position where you'll be able to try out the examples we provide later on, and use them as a starting point for your actual Hibernate projects.

If you're new to Ant, it wouldn't be a bad idea to read the manual a little bit to get a sense of how it works and what it can do for you; this will help make sense of the *build.xml* files we start working with in our examples. If you decide (or already know) you like Ant, and want to dig deeper, you can pick up O'Reilly's *Ant: The Definitive Guide* (after you finish *this* book, of course)!

What about...

...Eclipse, JBuilder, Sun ONE Studio (Forte for Java), or some other Java IDE? Well, you can certainly use these, but you're on your own as far as what you need to do to get Ant integrated into the build process. (Several already use Ant, so you might be starting out ahead; for the others you might have to jump through some hoops.) If all else fails, you can use the IDE to develop your own code, but invoke Ant from the command line when you need to use one of our build scripts.

Getting the HSQLDB Database Engine

Hibernate works with a great many relational databases; chances are, it will work with the one you are planning to use for your next project. We need to pick one to focus on in our examples, and luckily there's an obvious choice. The free, open source, 100% Java HSQLDB project is powerful enough that it forms the backing storage for several of our commercial software projects. Surprisingly, it's also incredibly self-contained and simple to install, so it's perfect to discuss here. (If you've heard of Hyper-sonicSQL, this is its current incarnation. Much of the Hibernate documentation uses the older name.)

TIP

Don't panic if you end up at <http://hsqldb.sourceforge.net/> and it seems like the project has been shut down. That's the wrong address—it's talking about the predecessor to the current HSQLDB project. Use the address below to find the current version of the database engine.

Why do I care?

Examples based on a database that everyone can download and easily experiment with mean you won't have to translate any of the SQL dialects or operating system commands to work with your available databases (and may even mean you can save a day or two learning how to download, install, and configure one of the more typical database environments). Finally, if hsqldb is new to you, chances are good you'll be impressed and intrigued, and may well end up using it in your own projects. As it says on the project home page (at <http://hsqldb.sourceforge.net>):

hsqldb is a relational database engine written in Java, with a JDBC driver, supporting a rich subset of ANSI-92 SQL (BNF tree format). It offers a small (less than 160k), fast database engine which offers both in memory and disk based tables. Embedded and server modes are available. Additionally, it includes tools such as a minimal web server, in-memory query and management tools (can be run as applets), and a number of demonstration examples.

*Go on, download
HSQLDB. Heck,
take two, they're
small!*

How do I do that?

Getting the database is simply a matter of visiting the project page at <http://hsqldb.sourceforge.net> and clicking the link to download the current stable version. This will take you to a typical SourceForge downloads page with the current release highlighted. Pick your mirror and download the zip archive. There's nothing to install or configure; we'll show you how to use it shortly.

What about...

...MySQL, PostgreSQL, Oracle, DB2, Sybase, Informix, or some other common database? Don't worry, Hibernate can work with all these and others. We'll talk about how you specify "dialects" for different databases later on. And if you really want, you can try to figure out how to work with your favorite from the start, but it will mean extra work for you in following along with the examples, and you'll miss out on a great opportunity to discover HSQLDB.

Getting Hibernate

This doesn't need much motivation! You picked up this book because you wanted to learn how to use Hibernate.

How do I do that?

Go to the Hibernate home page, <http://www.hibernate.org/>, and click on the “Download” link. The Binary Releases section will tell you which version is recommended for downloading; follow that advice. Make a note of the version you want and proceed to the “Download: SourceForge” link. It takes you to a SourceForge downloads page. Scroll down until you find the recommended release version of Hibernate itself (which will look something like *hibernate-2.x.y.zip* or *hibernate-2.x.y.tar.gz*). Choose the archive format that is most convenient for you and download it.

Pick a place that is suitable for keeping such items around, and expand the archive. We will use part of it in the next step, and investigate more of it later on. You may also want to poke around in there some yourself.

While you’re on the Hibernate downloads page, also pick up the Hibernate Extensions. They contain several useful tools which aren’t necessary for an application running Hibernate, but are very helpful for developers creating such applications. We’ll be using one to generate Java code for our first Hibernate experiment in the next chapter. This file-name will look like *hibernate-extensions-2.x.y.zip* (it won’t necessarily have the same version as Hibernate itself). Once again, pick your favorite archive format, download this file, and expand it next to where you put Hibernate.

Setting Up a Project Hierarchy

Although we’re going to start small, once we start designing data structures and building Java classes and database tables that represent them, along with all the configuration and control files to glue them together and make useful things happen, we’re going to end up with a lot of files. So let’s start out with a good organization from the beginning. As you’ll see in this process, between the tools you’ve downloaded and their supporting libraries, there are already a significant number of files to organize.

Why do I care?

If you end up building something cool by following the examples in this book, and want to turn it into a real application, you’ll be in good shape from the beginning. More to the point, if you set things up the way we describe here, the commands and instructions we give you throughout the examples will make sense and actually work. Many examples also