

#### Research Centers

Core Java  
Enterprise Java  
Mobile Java  
Tools & Methods  
JavaWorld Archives

#### Site Resources

Featured Articles  
News & Views  
Community  
Java Q&A  
JW Blogs  
Podcasts  
Site Map  
Careers  
New sletters  
Whitepapers  
RSS Feeds

#### About JavaWorld

Advertise  
Write for JW

## iBATIS, Hibernate, and JPA: Which is right for you?

### Object-relational mapping solutions compared

By K. L. Nitin, Ananya S., Mahalakshmi K., and S. Sangeetha, JavaWorld.com, 07/15/08

Print Feedback 24 Comments Like

Page 4 of 7

#### Hibernate:

Hibernate is an open source, lightweight object-relational mapping solution. The main feature of Hibernate is its support for object-based modeling, which allows it to provide a transparent mechanism for persistence. It uses XML to map a database to an application and supports fine-grained objects. The current version of Hibernate is 3.x, and it supports Java annotations and hence satisfies the EJB specification.

Hibernate includes

a very powerful query language called *Hibernate Query Language*, or HQL. HQL is very similar to SQL, and also defines some

additional conventions. HQL is completely object-oriented, enabling you to leverage the complete strength of the object-oriented pillars of inheritance, polymorphism, and association. HQL queries are case insensitive, except for the names of the Java classes and properties being used. HQL returns query results as objects that can be directly accessed and manipulated by the programmer. HQL also supports many advanced features of pagination and dynamic profiling that SQL has never supported. HQL does not require any explicit joins when working with multiple tables.

#### Why do we need Hibernate?

Entity beans, which have traditionally been used for object-relational mapping, are very difficult to understand and hard to maintain. Hibernate makes object-relational mapping simple by mapping the metadata in an XML file that defines the table in the database that needs to be mapped to a particular class. In other persistence frameworks, you need to modify the application class to achieve object-relational mapping; this is not necessary in Hibernate.

With Hibernate, you needn't worry about database changes, as manual changes in the SQL script files are avoided. If you ever need to change the database your application uses, that can be easily accommodated by altering the `dialect` property in the configuration file. Hibernate gives you the complete power of SQL, something that was never offered by earlier commercial ORM frameworks. Hibernate also supports many databases, including MySQL, Oracle, Sybase, Derby, and PostgreSQL, and works well with plain old Java object (POJO)-based models, too.

Hibernate generates JDBC code based on the underlying database chosen and so saves you the trouble of writing JDBC

#### JW's Most Read

Scala on their minds -- bloggers debate Scala's complexity  
Simplify the data access layer with Spring 3.1 and Java generics  
Bitcoin for beginners, Part 2: Bitcoin as a technology and network  
Why Yammer dropped Scala for Java  
Engine Yard vs Heroku: Ruby clouds (for Java too)

## Application Performance Management



APP SERVERS,  
SERVERS, DATABASES,  
CLOUD SERVICES, SAP,  
VIRTUAL SERVERS, etc ..

ManageEngine  
Applications Manager

DOWNLOAD  
FREE TRIAL

#### Featured White Papers

Observation Driven Testing: What else is your code doing?  
Software Agitation: Your Own Personal Code Reviewer  
Early & Often: Avoiding Security Flaws with CI, High Coverage

#### Newsletter sign-up [View all new sletters](#)

#### Enterprise Java Newsletter

Stay up to date on the latest tutorials and Java community news posted on JavaWorld

#### Sponsored Links

Slow Apps? Overloaded DBMS? - FREE Download  
FREE Download Feb 14 - March 7th Eliminate bottlenecks & maximize performance ScaleOut StateServer®

#### Sponsored Links

Optimize with a SATA RAID Storage Solution  
Range of capacities as low as \$1250 per TB.  
Ideal if you currently rely on servers/disks/JBODs

### Hibernate architecture

Internally, Hibernate uses JDBC, which provides a layer of abstraction to the database, while it employs the Java Transaction API (JTA) and JNDI to integrate with other applications. The connection information that the Hibernate needs to interact with the database is provided by the JDBC connection pool, which has to be configured.

Hibernate's architecture consists mainly of two interfaces -- `Session` and `Transaction` -- along with the `Query` interface, which is in the persistence layer of the application. The classes that are defined in the business layer of the application interact through independent metadata with the Hibernate persistence layer, which in turn talks to the database layer using certain JDBC APIs. In addition, Hibernate uses other interfaces for configuration, mainly the aptly named `Configuration` class. Hibernate also makes use of callback interfaces and some optional interfaces for extending the mapping functionality. The overall Hibernate architecture is illustrated in Figure 3.

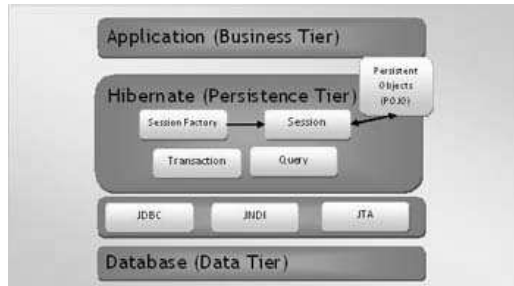


Figure 3. Hibernate architecture: The big picture

The major programming interfaces that are part of Hibernate are:

`org.hibernate.SessionFactory` is basically used to obtain a session instance, and can be seen as an analogue to the connection pooling mechanism. This is thread safe, as all the application threads can use a single `SessionFactory` (as long as Hibernate uses a single database). This interface is configured through the configuration file, which determines the mapping file to be loaded.

`org.hibernate.Session` provides a single thread that determines the conversation between the application and the database. This is analogous to a specific (single) connection. It is very lightweight and not thread safe.

`org.hibernate.Transaction` provides a single-thread object that spans through the application and determines an atomic unit of work. It basically abstracts JDBC, JTA, and CORBA transactions.

`org.hibernate.Query` is used to perform a query, either in HQL or in the SQL dialect of the underlying database. A `Query` instance is lightweight, and it is important to note that it cannot be used outside the session through which it was created.

### Configuring Hibernate

You configure Hibernate through an XML file named `hibernate.cfg.xml`. The configuration file aids in establishing a connection to a particular relational database. The configuration file should know which mapping file it needs to refer to. At runtime, Hibernate reads the mapping file and then uses it to build a dynamic Java class corresponding to that table of the database. A sample configuration file is shown in Listing 6.

Listing 6. `hibernate.cfg.xml`

```
<hibernate-configuration>
  <session-factory>
    <!-- local connection properties -->
    <property name="hibernate.connection.url">
      jdbc:mysql://localhost/hibernateDemo
    </property>
    <property name="hibernate.connection.driver_class">
      com.mysql.jdbc.Driver
    </property>
    <property name="hibernate.connection.username">
      root
    </property>
    <property name="hibernate.connection.password">
      infosys
    </property>
    <!-- dialect for MySQL -->
    <property name="dialect">
      org.hibernate.dialect.MySQLDialect
    </property>
    <property name="hibernate.show_sql">false</property>
    <property name="hibernate.transaction.factory_class">
      org.hibernate.transaction.JDBCTransactionFactory
    </property>
  </session-factory>
</hibernate-configuration>
```

Print

Feedback

Like



Like

Comments powered by

Our Commenting Network | Policies | Privacy


Add New Comment

Login




Showing 20 of 24 comments

Sort by newest first



Ahamed Firzadh

Hats off gr8 article.... Thanks for the detailed explanation which is also easy on a newbie :)


11/21/2011 06:14 AM 

Like Reply



Anonymous

IBatis, Hibernate, JPA  
Thank you for putting this together. It provides a very good introduction for me into ORM and persistence. Much appreciated.

06/26/2010 04:39 PM 

Like Reply



Anonymous

Thanks!!!  
Thanks a lot. Very useful article.


05/30/2010 05:34 AM 

Like Reply




Anonymous

Buen inicio  
Muchas gracias por el artículo, me dió lo necesario para empezar :)

05/22/2010 11:06 PM 

Like Reply




Anonymous

good article.  
Thanks

05/19/2010 02:14 AM 

Like Reply



Anonymous

Thanks!  
Great article!

04/20/2010 04:10 AM 

Like Reply



Anonymous

Nice comparison  
I was searching for a comparison between Hibernate and JPA. Thanks for the effort.

Sunil



Anonymous

Hibernate Vs IBatis  
The article is awesome for the beginners. Thanks for the valuable information that you have put on the Java World.

03/24/2010 02:21 AM

Like   Reply



Anonymous

great article and excellent explanation.

03/13/2010 09:21 AM

Like   Reply



Anonymous

Question  
The article mentions that JPA is non-portable. But wouldn't it be possible to use JPA in Rails using JRuby?

01/30/2010 08:51 AM

Like   Reply



Anonymous

Yes, It's a grate  
Can i say that, If we want to separate Business layer and database layer that time, we can use this method....

01/22/2010 02:03 AM

Like   Reply



Anonymous

Hibernate Vs IBatis  
IBatis is considered to be the way if you want to have full control over your queries + don't want to clutter your code with sqls.

I would contest that by saying hibernate provides the exact same feature with the concept of named queries if you ever needed that feature.

The only reason i think people tend to tilt towards IBatis is that it does not overwhelm you with tons of features which first time users find a bit daunting and get scared away.

01/21/2010 01:04 PM

Like   Reply



Anonymous

This is it.  
I was searching for database layer solutions. This article helped me a lot. I am enlightened. Thank you so much.

12/06/2009 09:45 PM

Like   Reply



Anonymous

Control over queries...  
"Hibernate provides a complete ORM solution, but offers you no control over the queries"

Dude, Session.createSQLQuery lets you use native sql, if that is not "full control over queries" then iBatis does not have it either! PD: RTFM

10/08/2009 02:36 PM

Like   Reply



Anonymous

Hibernate also lets you custom code sql mappings for its ORM use just like ibatis does too. so if you want/need to code your own sql you can!  
you can also configure it to use stored procs too.  
ref <http://docs.jboss.org/hibernat...>

03/24/2010 06:13 PM   in reply to Anonymous

Like   Reply



Anonymous

Question for the authors

iBATIS, Hibernate, and JPA: Which is right for you? - JavaWorld

application, that's why I choose iBATIS. On the other hand, I'll have control over both the Java and the DB; but no time to migrate the Stored Procedures to Java.

Performance will be a very important issue, as it's an application which will manage loads of users and heavy binary and xml stored data. I can tune the queries on iBATIS, but the cache mechanism provided by Hibernate may be a better option?

Thanks a lot!

10/06/2009 08:21 AM

Like Reply



Anonymous

iBatis is my choice - from experience...  
There's no silver bullet, so use the technology you are most familiar with or standard in the environment. BUT, having used iBatis and hibernate extensively, I always default to iBatis. I have spent too many hours trying to figure out what hibernate is doing - as performance can be an issue. I rarely work on a project that needs to access more than one database type (mysql, oracle, sybase, ms sql server, postgres etc).

I have just finished a .Net desktop project with iBatis as the db access layer. It works a treat!

A large financial organisation performed extensive research on java tools for their global architecture team ... and iBatis came out on top for the db access layer.

So if its a new project, give iBatis a try.

09/02/2009 12:26 AM

Like Reply



Anonymous

Hibernate's Simplicity is "Good"? Bah...  
How on Earth did you conclude that Hibernate's simplicity is 'good'? Let me tell you why it should be downgraded to 'average' if not 'poor'. The official Hibernate book is huge, and the number of times we've had to research how to solve a problem on our project is ridiculous. Ramp up time is long for new teammates. The reason why it seems community support is so good for Hibernate is because we see so many posts of people have problems with the technology.

08/03/2009 11:34 PM

Like Reply



Anonymous

it is a very helpful article and also has a very simple manner of telling. I was not bored while reading.  
Thank you so much..

07/17/2009 12:39 AM

Like Reply



Anonymous

Hibarnate seems to be the best  
Hibernate seems to be winner

07/09/2009 05:28 AM 1 Like

Like Reply

✉ Subscribe by email



Load more comments

Archived Discussions (Read only)

Forum migration complete *By Athen*

Forum migration update *By Athen*

Resources

- Learn more about the three technologies discussed in this article from the project homepages:
  - Hibernate
  - iBATIS

iBATIS, Hibernate, and JPA: Which is right for you? - JavaWorld

Java Persistence with Hibernate (Christian Bauer and Gavin King; Manning, November 2006) is the updated second edition of *Hibernate in Action*. Also see iBATIS in Action (Clinton Begin, Brandon Goodin, and Larry Meadors, 2007).

For broader coverage of Java persistence solutions including JDBC, OpenJPA, and pureQuery, see Persistence in the Enterprise: A Guide to Persistence Technologies (Roland Barcia, Geoffrey Hambrick, Kyle Brown, Robert Peterson, Kulvir Singh Bhogal; IBM Press, May 2008).

Ted Neward introduces the so-called object-relational impedance mismatch in his blog post "The Vietnam of computer science."

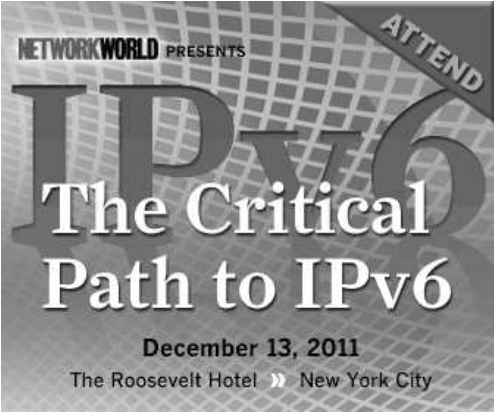
"Flexible reporting with JasperReports and iBatis" (Scott Monahan, JavaWorld, December 2007) is a hands-on introduction to the iBatis Data Mapper framework.

"Understanding the Java Persistence API" (Aditi Das, JavaWorld, January 2008) is a two-part introduction to Java-platform persistence with OpenJPA.

Java-source.net lists a roundup of open source persistence frameworks for Java.

Visit the JavaWorld Java Enterprise Edition research center for more articles about enterprise data management and Java persistence solutions.

Also see Network World's IT Buyer's Guides: Side-by-side comparison of hundreds of products in over 70 categories.



Sponsored Links

Web hosting reviews for you to choose a better host  
ManageEngine: End-to-End Java Performance Management. Download Product Now!

RESEARCH CENTERS	IDG Network			
Core Java	CFOworld	GamePro	IDG Ventures	Macworld
Enterprise Java	CIO	Games.net	InfoWorld	Network World
Mobile Java	Computerworld	IDG Connect	ITworld	PC World
Tools & Methods	CSO	IDG Knowledge Hub	JavaWorld	
JavaWorld Archives	DEMO	IDG TechNetwork	LinuxWorld	

About Us | Advertise | Contact Us | Terms of Service/Privacy | AdChoices

Copyright, 2006-2011 Infoworld, Inc. All rights reserved.