

OBJECT AND RELATIONAL MAPPING (ORM) WITH HIBERNATE

Hibernate is a high-performance object/relational persistence and query service. The most flexible and powerful object/relational solution on the market, Hibernate takes care of the mapping from Java classes to database tables and from Java data types to SQL data types. It provides data query and retrieval facilities that significantly reduce development time. Hibernate's design goal is to relieve the developer from 95% of common data persistence-related programming tasks by eliminating the need for manual, hand-crafted data processing using SQL and JDBC.

Challenge: Working with both object-oriented software and relational databases can be cumbersome and time consuming.

Development costs are significantly higher due to a paradigm mismatch between how data is represented in objects versus relational databases. Many software developers and architects estimate that up to 30% of their code is needed to deal with this infrastructure concern. Hibernate directly addresses this challenge by providing the ability to map an object model's data representation to a relational data model and its corresponding database schema.

Hibernate offers object/relational mapping (ORM) for Java applications.

Hibernate provides the ability to map an object model's data representation to a relational data model and its corresponding database schema. Hibernate is licensed under the open source GNU Lesser General Public License (LGPL) and is free to download and free to use for both development and production deployments. The license also allows ISVs to embed and distribute Hibernate free of charge.

Hibernate eliminates repetitive and tedious coding work and enables developers to focus on the business problem. No matter which application development strategy you prefer—starting top-down with a new business problem and model or bottom up with an existing database schema—Hibernate can significantly reduce development time.

INCREASED DEVELOPER PRODUCTIVITY AND REDUCED MAINTENANCE COSTS

Hibernate also substantially reduces lines of code by automating ORM. Hibernate provides a buffer between the two data representations and enables a more elegant use of object orientation on the Java side—all while keeping the relational schema normalized and guaranteeing data integrity.

Optimized performance

By allowing Hibernate to centrally handle persistence, optimizations can be applied across the board all of the time—unlike hand-coding persistence. For example, a modular two-layer caching system can be configured using JBoss Cache. This would allow the application to benefit from across the board caching, rather than isolated, hand-written performance enhancements.

Standards-driven development

Hibernate is typically used in Java Swing applications, Java Servlet-based applications, or J2EE applications using Enterprise Java Bean (EJB) session beans. Hibernate 3.0 plays a critical role in the EJB 3.0 Java Persistence API which brings a dramatically simplified persistence model to the EJB standard. By combining Hibernate 3 with Hibernate Annotations, developers can achieve an EJB 3.0 style of programming outside of the EJB 3.0 container and within standalone Java applications.

Enterprise-class reliability and scalability

Hibernate is well known for its excellent stability and quality—proven by the acceptance and use by tens of thousands of Java developers. Hibernate was designed to work in an application server cluster and deliver a highly scalable architecture. Hibernate scales well in any environment: Use it to drive your in-house Intranet that serves hundreds of users or for mission-critical applications that serve hundreds of thousands.

Transparent persistence

Hibernate requires no interfaces or base classes for persistent classes and enables any class or data structure to be persistent. Furthermore, Hibernate enables faster build procedures since it does not introduce build-time source or byte code generation or processing.

Flexible object/relational mapping and simple APIs

Hibernate is driven by XML mapping documents that define the ORM and generate database table and constraint creation scripts. Hibernate supports a variety of inheritance mapping strategies and all entity association mapping styles—including one-to-many, one-to-one, and many-to-many. Hibernate also supports bi-directional and unidirectional associations as well as fine-grained composition for dependent value objects.

Hibernate includes a Core API for application code, an Extension API for customizations, and a Metadata API for applications that require access to persistence metadata (to handle runtime mapping creation and changes, for example). The APIs in Hibernate 3 provide a superset of the functionality of EJB 3.0 Java Persistence, which means you can use EJB 3.0 Entity Manager and Query APIs with Hibernate when used alone or within the version of JBoss Application Server that supports EJB 3.0.

Object-oriented query language

Hibernate provides a powerful query language (HQL) that is expressed in a familiar SQL-like syntax and includes full support for polymorphic queries. Queries may also be expressed in the native SQL dialect of your database or with powerful programmatic and type-safe Criteria and Example queries.

Operate in managed and non-managed environments

Hibernate can operate within any J2EE application server—including JBoss Application Server—and can be configured and managed via JMX (Java Management Extension) MBeans. Hibernate's transaction system integrates with J2EE application server transactions via JTA

(Java Transaction API). Hibernate can also run outside of an application server container by bypassing the time-consuming deployment step. This is also a useful feature for those interested in using Hibernate within standalone non-managed Java applications.

High-performance

Hibernate includes lazy initialization, outer join fetching, batch fetching, and union fetching, as well as support for optimistic locking with automatic versioning and time stamping. Hibernate requires no special database tables or fields and generates much of the SQL at system initialization time instead of runtime. Hibernate consistently offers superior performance over straight JDBC coding.

Dual-layer caching

Hibernate's dual-layer cache architecture delivers thread safety, non-blocking data access, session level cache, optional and pluggable second-level cache, and optional query cache. Hibernate also works well in distributed application environments where other applications have simultaneous access to the database.

SUPPORT SERVICES FROM THE SOURCE

JBoss pioneered the Professional Open Source model whereby open source software is backed with expert support services delivered by product experts. JBoss and the large network of JBoss Authorized Service Partners offer a comprehensive set of support offerings for Hibernate including professional support, consulting, and training.

Hibernate training and support

- **Professional Support** helps you to overcome all Hibernate-related issues –including bug and patch management, production support, and general Hibernate development and deployment assistance. Three different levels of support are available, ranging from 8x5 support with 48-hour response times to 24x7 support with 2-hour response times. All JBoss Professional Support packages offer unlimited support with no restrictions on the number of hours or calls.

- **JBoss Network** access is included with every Hibernate Professional Support contract. JBoss Network is a support offering that integrates knowledge access, software distribution, and application management.
- **Consulting** includes both on-site and remote consulting services delivered by a Hibernate expert. When in-depth product knowledge is essential, our consulting services deliver the personal attention directly from the source that is critical for ensuring a successful launch.
- **Training** is delivered both through public courses hosted in major cities around the world or as private classes hosted on-site.
- Hibernate Training is targeted at experienced developers who wish to become experts on Hibernate. This two day course includes approximately 70% theory and 30% hands-on labs. A half-day online tutorial is available for those less familiar with Hibernate who wish to prepare themselves in advance for the intensive two-day course.

New for Hibernate 3

Hibernate 3's "Hibernate for the Enterprise" focus has built on the success of Hibernate 2 and extended it with enterprise-class functionality including:

- Support for EJB 3.0 Annotations, Entity Manager, and Java Persistence API
- ORM improvements that support virtualized filtering for temporal, historical, regional, and permissioned data
- Single object to multi-table mapping, bulk update and delete by query, and the ability to override generated SQL with hand-written SQL
- JMX-enabled statistics reporting and monitoring through any JMX console
- XML binding that enables data to be represented as XML and Java objects interchangeably
- Event-driven design that enables custom event objects to be created and registered to handle auditing scenarios or cascaded behavior semantics

MINIMUM SYSTEM REQUIREMENTS

- Hibernate 2 requires JDK 1.2 or higher (1.4.2 and 1.5.0 are recommended)
- Hibernate 3 requires JDK 1.3.1 or higher (1.4.2 and 1.5.0 are recommended)
- 128 MB RAM
- 50 MB hard disk space
- 400 MHz CPU

Supported operating systems

Hibernate is 100% pure Java and therefore interoperable with most operating systems that are capable of running a Java Virtual Machine (JVM); including Windows®, UNIX®, and Linux®.

Supported databases

Hibernate is interoperable with any JDBC-compliant database and supports more than 20 popular dialects of SQL out of the box including Oracle, DB2, Sybase, MS SQL Server, PostgreSQL, MySQL, HypersonicSQL, Mckoi SQL, SAP DB, Interbase, Pointbase, Progress, FrontBase, Ingres, Informix, and Firebird

Supported standards

- Java Management Extension (JMX) 1.2
- Java Transaction API (JTA) 1.0.1B
- JDBC 3.0
- Java Naming and Directory Interface (JNDI) 1.2.1
- EJB 3.0 (currently in public draft release)



JBoss SALES AND INQUIRIES

NORTH AMERICA
1-888-REDHAT1
jboss.com
sales@redhat.com

**EUROPE, MIDDLE EAST
AND AFRICA**
00800 7334 2835
europe.redhat.com/jboss
europe@redhat.com

ASIA PACIFIC
+65 6490 4200
apac.redhat.com/jboss
apac@redhat.com

LATIN AMERICA
+54 11 4329 7300
latam.redhat.com/jboss
info-latam@redhat.com

