

**CS 683 Emerging Technologies
Fall Semester, 2004
Doc 27 Hibernate Example
Contents**

References	2
Hibernate	3
First Example	3
Database	3
People Class.....	4
Mapping – People.hbm.xml	5
Sample Connection.....	6
Setup	9
Using Database Sequence & Hibernate Properties	10
Database	10
People Class	10
Mapping – People.hbm.xml	11
Sample Usage	12

Copyright ©, All rights reserved. 2004 SDSU & Roger Whitney,
5500 Campanile Drive, San Diego, CA 92182-7700 USA.
OpenContent (<http://www.opencontent.org/opl.shtml>) license
defines the copyright on this document.

References

Hibernate in Action, Bauer & King, Manning, 2005

Hibernate First Example

Storing Person objects in table

Database Table

id	first_name	last_name

SQL Used to Create Table

```
CREATE TABLE PEOPLE
(FIRST_NAME varchar(50) NULL ,
LAST_NAME varchar(50) NULL ,
ID int NOT NULL ,
PRIMARY KEY (id));
```

People Class

```
package sample;

public class People {
    String firstName;
    String lastName;
    long id;

    public People () {super();}

    public People(String first, String last) {
        firstName = first;
        lastName = last;
    }

    public String getLastName() { return lastName; }

    public String getFirstName() { return firstName; }

    public void setFirstName( String name) { firstName = name; }

    public void setLastName( String name) { lastName = name; }

    public long getId() { return id; }

    public void setId(long l) {id = l; }

    public String toString() {return firstName + " " + lastName + id;}
}
```

Mapping – People.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
  "-//Hibernate/Hibernate Mapping DTD//EN"
  "http://hibernate.sourceforge.net/hibernate-mapping-2.0.dtd" >

<hibernate-mapping package="sample">
  <class
    name="People"
    table="people" >
    <id
      name="id"
      type="java.lang.Long"
      column="id" >
      <generator class="assigned"/>
    </id>

    <property
      name="lastName"
      column="last_name"
      type="string"
      not-null="false"
      length="50" />
    <property
      name="firstName"
      column="first_name"
      type="string"
      not-null="false"
      length="50" />
  </class>
</hibernate-mapping>
```

Sample Connection

```
package sample;

import java.sql.Connection;
import java.sql.DriverManager;
import java.util.List;

import net.sf.hibernate.HibernateException;
import net.sf.hibernate.MappingException;
import net.sf.hibernate.Query;
import net.sf.hibernate.Session;
import net.sf.hibernate.SessionFactory;
import net.sf.hibernate.Transaction;
import net.sf.hibernate.cfg.Configuration;

public class Main {
    public static void main(String[] args) throws Exception {
        sampleRead();
        sampleWrite();
    }

    public static Connection getJdbcConnection() throws Exception {
        String dbUrl = "jdbc:postgresql://localhost/glorpTests";
        String user = "whitney";
        String password = "secret";
        System.out.println("Load Driver!");
        Class.forName("org.postgresql.Driver");
        return DriverManager.getConnection( dbUrl, user, password);
    }
}
```

Sample Connection Continued

```
static Session getHibernateSession() throws  
    MappingException, HibernateException, Exception {  
  
    Configuration config = new Configuration();  
    config.addClass(sample.People.class);  
    config.setProperties(System.getProperties());  
  
    SessionFactory sessions = config.buildSessionFactory();  
    Session session = sessions.openSession(getJdbcConnection());  
    return session;  
}
```

```
static void sampleWrite() throws  
    MappingException, HibernateException, Exception {  
    Session session = getHibernateSession();  
    Transaction save = session.beginTransaction();  
    People newPerson = new People("Jack", "Frost");  
    newPerson.setId(1);  
    session.save(newPerson);  
    newPerson = new People("Jack", "Ripper");  
    newPerson.setId(2);  
    session.save(newPerson);  
    save.commit();  
    session.close();  
}
```

Sample Connection Continued

```
static void sampleRead() throws  
    MappingException, HibernateException, Exception {  
    Session session = getHibernateSession();  
    Query getByName =  
        session.createQuery(  
            "from People p where p.lastName = :var");  
    getByName.setString("var", "Frost");  
    List result = getByName.list();  
    System.out.println("Number of Objects: " + result.size());  
    People frost = (People) result.get( 0);  
    System.out.println(frost);  
    session.close();  
}  
}
```

Setup

Directory sample contains:

- People.java
- Main.java
- People.hbm.xml

Class path contains:

- Directory containing sample
- Hibernate/hibernate2.jar
- Hibernate/lib/*.jar
- Hibernate/etc/Hibernate.properties

Where Hibernate is the location of your Hibernate installation.

Using Database Sequence & Hibernate Properties

Database Table

<code>id</code>	<code>first_name</code>	<code>last_name</code>

SQL Used to Create Table

```
CREATE TABLE PEOPLE
(FIRST_NAME varchar(50) NULL ,
LAST_NAME varchar(50) NULL ,
ID serial NOT NULL ,
CONSTRAINT PEOPLE_PK PRIMARY KEY (id),
CONSTRAINT PEOPLE_UNIQ UNIQUE (id))
```

People Class

Same as before

Mapping – People.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
  "-//Hibernate/Hibernate Mapping DTD//EN"
  "http://hibernate.sourceforge.net/hibernate-mapping-2.0.dtd" >

<hibernate-mapping package="sample">
  <class
    name="People"
    table="people" >
    <id
      name="id"
      type="java.lang.Long"
      column="id" >
      <generator class="increment"/>  <!-- Changed here -->
    </id>

    <property
      name="lastName"
      column="last_name"
      type="string"
      not-null="false"
      length="50" />
    <property
      name="firstName"
      column="first_name"
      type="string"
      not-null="false"
      length="50" />
  </class>
</hibernate-mapping>
```

Sample Usage

Only changed methods from Main are shown

```
public static void main(String[] args) throws Exception {  
    sampleWrite();  
    sampleUpdate();  
    sampleRead();  
}  
  
static void sampleWrite() throws MappingException, HibernateException,  
    Exception {  
    Session session = getHibernateSession();  
    Transaction save = session.beginTransaction();  
    People newPerson = new People("Roger", "Whitney");  
    session.save(newPerson);      //Id is set automatically  
    save.commit();  
    session.close();  
}  
  
static void sampleUpdate() throws MappingException,  
    HibernateException,  
    Exception {  
    Session session = getHibernateSession();  
    System.out.println(session.isConnected());  
    Transaction update = session.beginTransaction();  
    Query getFrost =  
        session.createQuery("from People p where p.lastName = :var");  
    getFrost.setString("var", "Frost");  
    List result = getFrost.list();  
    People frost = (People) result.get(0);  
    frost.setFirstName("Roger");  
    update.commit();  
    session.close();  
}
```

```
static void sampleRead() throws MappingException, HibernateException,  
    Exception {  
    Session session = getHibernateSession();  
    Query getFrost = session.createQuery(  
        "from People p where p.lastName = :var1 and p.firstName = :var2"  
    );  
    getFrost.setString("var1", "Whitney");  
    getFrost.setString("var2", "Roger");  
    List result = getFrost.list();  
    System.out.println("Number of Objects: " + result.size());  
    System.out.println(result.get(0));  
    session.close();  
}
```

```
static void sampleUpdate() throws MappingException, HibernateException,  
    Exception {  
    Session session = getHibernateSession();  
    System.out.println(session.isConnected());  
    Transaction update = session.beginTransaction();  
    Query getFrost =  
        session.createQuery("from People p where p.lastName = :var");  
    getFrost.setString("var", "Frost");  
    List result = getFrost.list();  
    People frost = (People) result.get(0);  
    frost.setFirstName("Roger");  
    update.commit();  
    session.close();  
}
```

```
static Session getHibernateSession() throws MappingException,  
HibernateException, Exception {  
    Configuration config = new Configuration();  
    config.addClass(sample.People.class);  
    config.setProperties(System.getProperties());  
  
    SessionFactory sessions = config.buildSessionFactory();  
    Session session = sessions.openSession();  
    return session;  
}
```

Hibernate/etc/Hibernate.properties

PostgreSQL

```
hibernate.dialect net.sf.hibernate.dialect.PostgreSQLDialect  
hibernate.connection.driver_class org.postgresql.Driver  
hibernate.connection.url jdbc:postgresql:glorpTests  
hibernate.connection.username whitney  
hibernate.connection.password secret  
hibernate.query.substitutions yes 'Y'
```

#####

C3P0 Connection Pool###

#####

```
hibernate.connection.username=whitney  
hibernate.connection.password=secret  
hibernate.connection.url=jdbc:postgresql://localhost/glorpTests  
hibernate.connection.driver_class=org.postgresql.Driver  
hibernate.dialect=net.sf.hibernate.dialect.PostgreSQLDialect  
hibernate.c3p0.max_size 2  
hibernate.c3p0.min_size 2  
hibernate.c3p0.timeout 5000  
hibernate.c3p0.max_statements 100  
hibernate.c3p0.idle_test_period 3000  
hibernate.c3p0.acquire_increment 2  
hibernate.c3p0.validate false
```

Just the changes from the default are shown

Hibernate/etc/log4j.properties

```
### direct log messages to stdout ###
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{ABSOLUTE} %5p
%c{1}:%L - %m%n

### direct messages to file hibernate.log #####
#log4j.appender.file=org.apache.log4j.FileAppender
#log4j.appender.file.File=hibernate.log
#log4j.appender.file.layout=org.apache.log4j.PatternLayout
#log4j.appender.file.layout.ConversionPattern=%d{ABSOLUTE} %5p
%c{1}:%L - %m%n

### set log levels - for more verbose logging change 'info' to 'debug' #####
log4j.rootLogger=warn, stdout

log4j.logger.net.sf.hibernate=info

### log just the SQL
log4j.logger.net.sf.hibernate.SQL=debug

### log JDBC bind parameters ####
log4j.logger.net.sf.hibernate.type=info

### log schema export/update ####
log4j.logger.net.sf.hibernate.tool.hbm2ddl=debug

### log cache activity ####
log4j.logger.net.sf.hibernate.cache=debug

### enable the following line if you want to track down connection #####
### leakages when using DriverManagerConnectionProvider #####
#log4j.logger.net.sf.hibernate.connection.DriverManagerConnectionProvider
=trace
```

Setup

Directory sample contains:

- People.java
- Main.java
- People.hbm.xml

Class path contains:

- Directory containing sample
- Hibernate/hibernate2.jar
- Hibernate/lib/*.jar

Properties

- Hibernate/etc/Hibernate.properties
- Hibernate/etc/log4j.properties

`java -DpropertyFile`

Where Hibernate is the location of your Hibernate installation.