

## 使用 Xdoclet 和 Ant 构建 Hibernate 映射和配置文件

### 功能描述:

在 ssh 项目中, 你是否存在着这样一种困惑, 那就是需要手动去配置 Hibernate 的映射文件和配置文件。这个过程是相当痛苦的, 需要写一大堆的 xml 文件。Xdoclet 就是为了简化这一工作的。它基于 Ant 或 Maven, 可以简化你的工作, 使你在使用 Hibernate 时, 不用在手动去为每个 pojo 类写对应的映射文件。当然你可以使用 Hibernate 的注解完成同样的工作, 但具体在 ssh 的项目中到底是选择注解方式还是选择配置方式, 这完全取决于你自己或项目经理。关于是否采用注解还是配置文件的来方式来完成 Hibernate 的映射, 这个在网上有着很大的争议。这里我不做过多评价。个人更加倾向于使用配置文件。因为配置文件比较灵活, 修改后不用重新编译成 Java 字节码文件, 相反使用注解的话你需要及时的去编译。好了, 话不多说, 我们进入正题。

### 环境描述:

|               |                                                                                                                   |
|---------------|-------------------------------------------------------------------------------------------------------------------|
| Eclipse3.6    | <a href="http://eclipse.org">http://eclipse.org</a>                                                               |
| Ant1.7.1      | <a href="http://ant.apache.org/">http://ant.apache.org/</a>                                                       |
| Xdoclet-1.2.3 | <a href="http://xdoclet.sourceforge.net/xdoclet/index.html">http://xdoclet.sourceforge.net/xdoclet/index.html</a> |
| Mysql5.0.22   | <a href="http://www.mysql.com/">http://www.mysql.com/</a>                                                         |
| Hibernate3.X  | <a href="http://www.hibernate.org/">http://www.hibernate.org/</a>                                                 |
| WindowsXp sp3 |                                                                                                                   |

温馨提示: 这里重点需要说明一下此处 Xdoclet 版本为 1.2.3, 网上有一些关于 Xdoclet 的配置是基于 1.0.4 以前的版本。这两个版本之间存在着较大的区别, 主要体现在 Ant 的构建文件 build.xml 中! 请确保你的 Xdoclet 版本为 1.2.3。

Eclipse 的安装就不介绍了。

### Ant 安装:

将下载下来的 Ant 解压到一个目录(建议该目录不含空格), 笔者解压到

E:\JAR\apache-ant-1.7.1

Ant 解压后的目录结构为:



### 设置环境变量:

要是 Ant 能够正常使用, 我们需要设置环境变量(通常这是在 Windows 下)

设置 ANT\_HOME

修改 Path



ANT\_HOME

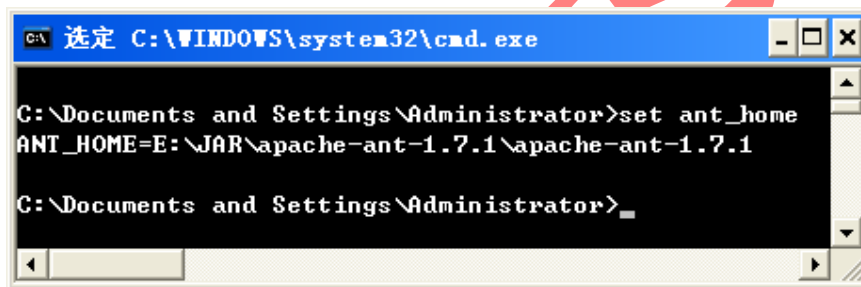


Path

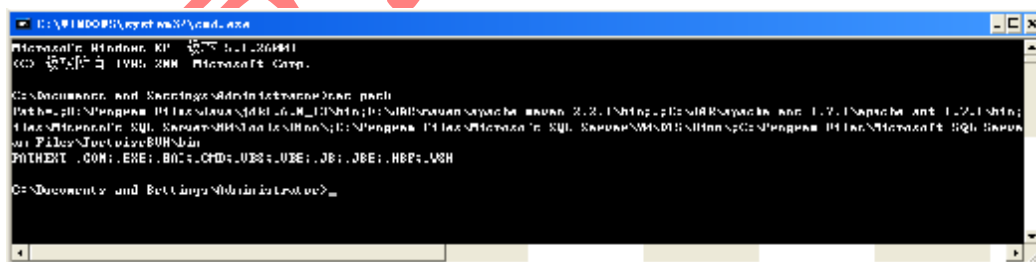
查看环境变量是否正确

打开命令行窗口

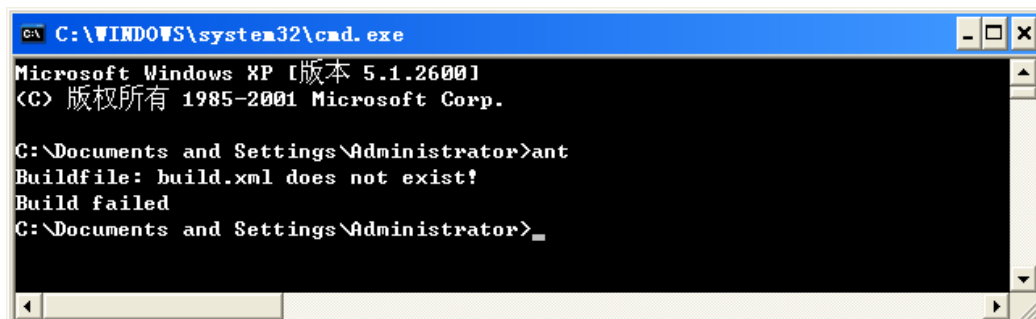
输入 set ANT\_HOME 查看 ANT\_HOME



输入 set Path 查看 Path



设置正确后在命令行下输入 ant



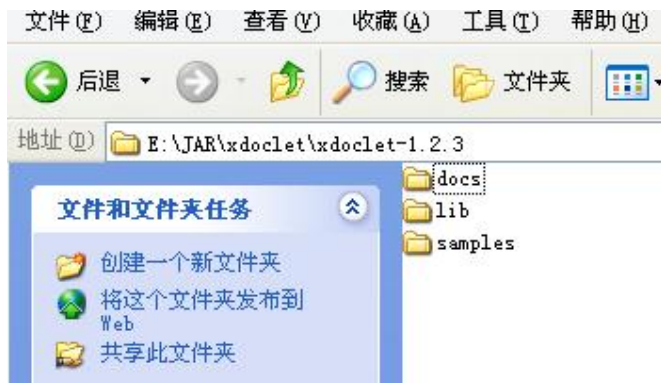
此时表示已将 ant 成功安装到操作系统中了,你可以使用 ant 的各种功能

Xdoclet-1.2.3 安装

解压 Xdoclet-1.2.3 文件

笔者这里解压到 E:\JAR\xdoclet\xdoclet-1.2.3

解压后的文件格式为:



这里需要记住 xdoclet 的解压路径 E:\JAR\xdoclet\xdoclet-1.2.3, 因为在 Ant 构建脚本中需要用到它

Mysql 的安装不用介绍了,相信大家都不陌生。

准备 Hibernate 依赖的 jar

|                            |          |                    |                 |
|----------------------------|----------|--------------------|-----------------|
| ant-1.6.5                  | 1,010 KB | Executable Jar ... | 2010-9-30 10:34 |
| ant-antlr-1.6.5            | 6 KB     | Executable Jar ... | 2010-9-30 10:34 |
| ant-junit-1.6.5            | 73 KB    | Executable Jar ... | 2010-9-30 10:34 |
| ant-launcher-1.6.5         | 9 KB     | Executable Jar ... | 2010-9-30 10:34 |
| antlr-2.7.6                | 434 KB   | Executable Jar ... | 2010-9-30 10:34 |
| ant-swing-1.6.5            | 7 KB     | Executable Jar ... | 2010-9-30 10:34 |
| asm                        | 26 KB    | Executable Jar ... | 2010-9-30 10:34 |
| asm-attrs                  | 17 KB    | Executable Jar ... | 2010-9-30 10:34 |
| c3p0-0.9.0                 | 476 KB   | Executable Jar ... | 2010-9-30 10:34 |
| cglib-2.1.3                | 276 KB   | Executable Jar ... | 2010-9-30 10:34 |
| checkstyle-all             | 1,177 KB | Executable Jar ... | 2010-9-30 10:34 |
| cleanimports               | 140 KB   | Executable Jar ... | 2010-9-30 10:34 |
| commons-collections-2.1.1  | 172 KB   | Executable Jar ... | 2010-9-30 10:34 |
| commons-logging-1.0.4      | 38 KB    | Executable Jar ... | 2010-9-30 10:34 |
| concurrent-1.3.2           | 168 KB   | Executable Jar ... | 2010-9-30 10:34 |
| connector                  | 18 KB    | Executable Jar ... | 2010-9-30 10:34 |
| dom4j-1.6.1                | 307 KB   | Executable Jar ... | 2010-9-30 10:34 |
| ehcache-1.2                | 116 KB   | Executable Jar ... | 2010-9-30 10:34 |
| hibernate3                 | 2,128 KB | Executable Jar ... | 2010-9-30 10:33 |
| jaas                       | 102 KB   | Executable Jar ... | 2010-9-30 10:34 |
| jacc-1.0-fr                | 24 KB    | Executable Jar ... | 2010-9-30 10:34 |
| javassist                  | 449 KB   | Executable Jar ... | 2010-9-30 10:34 |
| jaxen-1.1-beta-7           | 222 KB   | Executable Jar ... | 2010-9-30 10:34 |
| jboss-cache                | 506 KB   | Executable Jar ... | 2010-9-30 10:34 |
| jboss-common               | 570 KB   | Executable Jar ... | 2010-9-30 10:34 |
| jboss-jmx                  | 578 KB   | Executable Jar ... | 2010-9-30 10:34 |
| jboss-system               | 219 KB   | Executable Jar ... | 2010-9-30 10:34 |
| jdbc2_0-stdext             | 7 KB     | Executable Jar ... | 2010-9-30 10:34 |
| jgroups-2.2.8              | 1,574 KB | Executable Jar ... | 2010-9-30 10:34 |
| jta                        | 9 KB     | Executable Jar ... | 2010-9-30 10:34 |
| junit-3.8.1                | 119 KB   | Executable Jar ... | 2010-9-30 10:34 |
| log4j-1.2.11               | 343 KB   | Executable Jar ... | 2010-9-30 10:34 |
| mysql-connector-java-5.0.5 | 687 KB   | Executable Jar ... | 2010-9-30 10:37 |
| oscache-2.1                | 112 KB   | Executable Jar ... | 2010-9-30 10:34 |
| proxool-0.8.3              | 465 KB   | Executable Jar ... | 2010-9-30 10:34 |
| swarmcache-1.0rc2          | 30 KB    | Executable Jar ... | 2010-9-30 10:34 |
| syndiag2                   | 139 KB   | Executable Jar ... | 2010-9-30 10:34 |
| versioncheck               | 6 KB     | Executable Jar ... | 2010-9-30 10:34 |
| xerces-2.6.2               | 988 KB   | Executable Jar ... | 2010-9-30 10:34 |
| xml-apis                   | 121 KB   | Executable Jar ... | 2010-9-30 10:34 |

打开 eclipse, 创建一个 Java 工程, 名为 xdoclet

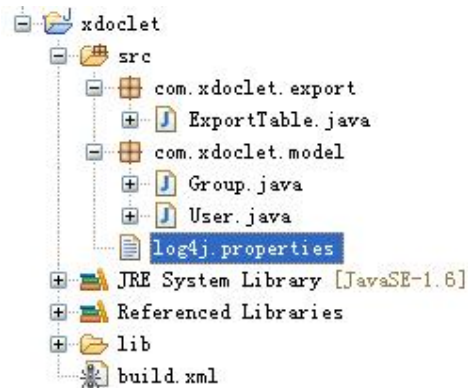
在工程上右键添加一个名为 build.xml 的文件

在工程上右键添加一个名为 lib 的文件夹, 拷贝 hibernate 依赖的 jar 包及 mysql 驱动, 最后将该 lib 文件夹设置到 classpath 路径下

两个实体类一个为 Group(组), 一个为(User)

为了更好的显示日志信息, 添加 log4j.properties 文件到 src 路径下

最后你看到的项目结构应该是这样的:



配置实体类

User.java

```
package com.xdoclet.model;

/**
 * @hibernate.class
 *     table="t_user"
 * @author welcome
 *
 */
public class User {
    private String userId;
    private String userName;
    private Group group;

    /**
     * @hibernate.id column="userId"
     * generator-class="assigned"
     */

    public String getUserId() {
        return userId;
    }

    public void setUserId(String userId) {
        this.userId = userId;
    }

    /**
     * @hibernate.property
     */
}
```

```
public String getUserName() {
    return userName;
}

public void setUserName(String userName) {
    this.userName = userName;
}

/**
 * @hibernate.many-to-one
 *     column="groupId"
 *     cascade="all"
 *     class="com.xdoclet.model.Group"
 * @param group
 */
public Group getGroup() {
    return group;
}

public void setGroup(Group group) {
    this.group = group;
}
}
```

Group.java

```
package com.xdoclet.model;
import java.util.Set;
/**
 * @hibernate.class
 *     table="t_group"
 * @author welcome
 */
public class Group {
    private String groupId;
    private String groupName;
    private Set userSets;
    /**
     * @hibernate.id
     *     column="groupId"
     *     generator-class="assigned"
     * @return
     */
    public String getGroupId() {
        return groupId;
    }

    public void setGroupId(String groupId) {
        this.groupId = groupId;
    }
}
```

```
}  
/**  
 * @hibernate.property  
 *     column="groupName"  
 * @return  
 */  
public String getGroupName() {  
    return groupName;  
}  
public void setGroupName(String groupName) {  
    this.groupName = groupName;  
}  
/**  
 * @hibernate.set inverse="true"  
 * @hibernate.collection-key column="groupId"  
 * @hibernate.collection-one-to-many  
 *     class="com.xdoclet.model.User"  
 * @return  
 */  
public Set getUserSets() {  
    return userSets;  
}  
public void setUserSets(Set userSets) {  
    this.userSets = userSets;  
}  
}
```

注意: 实体类中的注解, 是 xdoclet 中的, 这里不做解释, 具体可以去查看 xdoclet 关于 hibernate 的相关文档

<http://xdoclet.sourceforge.net/xdoclet/tags/hibernate-tags.html>

log4j.properties 配置文件详情

```
#All level less than INFO will be logged  
log4j.rootLogger=INFO, A1  
  
#A1 is the output device  
log4j.appender.A1=org.apache.log4j.FileAppender  
log4j.appender.A1.File=e:/log4j.htm  
  
#use html layout  
log4j.appender.A1.layout=org.apache.log4j.HTMLLayout
```

此文件会将日志信息记录为 html 格式的文件, 然后输出到 E 盘下, html 格式的日志信息看起来更加舒服。不信你可以去你目录下找找看。

最后我们来关注最重要的文件 build.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<project name="使用xdoclet映射hibernate" basedir=".">
```

```
<!-- 定义源文件目录变量 -->
<property name="src.dir" value="${basedir}/src" />
<!-- 定义xdoclet的目录 -->
<property name="xdoclet.home"
value="E:/JAR/xdoclet/xdoclet-1.2.3"></property>
<!-- 定义构建路径 -->
<path id="xdoclet.classpath">
    <fileset dir="${xdoclet.home}/lib">
        <include name="*.jar" />
    </fileset>
</path>

<path id="lib.classpath">
    <pathelement path="${java.class.path}" />
    <fileset dir="${xdoclet.home}/lib">
        <include name="**/*.jar" />
    </fileset>
    <fileset dir="${basedir}/lib">
        <include name="**/*.jar" />
    </fileset>
</path>
<!-- 生成Hibernate的映射文件 -->
<target name="生成Hibernate的映射文件"
unless="hibernatedoclet.unnecessary"
description="Generate Hibernate mapping files">
    <taskdef name="hibernatedoclet"
classname="xdoclet.modules.hibernate.HibernateDocletTask"
classpathref="xdoclet.classpath" />
    <hibernatedoclet destdir="${src.dir}" mergedir="${src.dir}"
excludedtags="@version,@author,@todo,@see"
verbose="false">
        <fileset dir="${src.dir}">
            <include name="com/xdoclet/model/*.java" />
        </fileset>
        <hibernate version="3.0" />
    </hibernatedoclet>
</target>
<!-- 生成Hibernate配置文件 -->
<target name="生成Hibernate配置文件" depends="生成Hibernate的映射文件">
    <taskdef name="hibernatedoclet"
classname="xdoclet.modules.hibernate.HibernateDocletTask"
classpathref="xdoclet.classpath" />
    <hibernatedoclet destdir="${src.dir}">
```

```
<fileset dir="${src.dir}">
    <include name="com/xdoclet/model/*.java" />
</fileset>
<hibernatecfg destDir="${src.dir}" version="3.0"
    hbm2ddl="create-update"
jdbcUrl="jdbc:mysql://localhost:3306/xdoclet"
    driver="com.mysql.jdbc.Driver" userName="root"
password="root"
    dialect="org.hibernate.dialect.MySQL5Dialect"
showSql="true">
    <otherProperty name="hbm2ddl" value="create-update" />
    <otherProperty name="format_sql" value="true" />
</hibernatecfg>
</hibernatedoclet>
</target>
<!-- 导出数据库表结构 -->
<target name="导出数据库表结构" depends="生成Hibernate配置文件">
    <taskdef name="schemaexport"
classname="org.hibernate.tool.hbm2ddl.SchemaExportTask"
        classpathref="lib.classpath"/>

    <property name="hibernate.dialect"
value="org.hibernate.dialect.MySQL5Dialect"/>
    <property name="hibernate.format_sql" value="true"/>
    <property name="hibernate.use_sql_comments" value="true"/>

    <schemaexport
        output="schema-export.sql"
        quiet="no"
        text="yes"
        drop="no"
        delimiter=";"
    >
    <fileset dir="${basedir}/src">
        <include name="com/xdoclet/model/*.hbm.xml" />
    </fileset>
    </schemaexport>
</target>

</project>
```

关于此文件不想做过多解释,细心的朋友可以自己去揣摩,前提是你要有 ant 的基础知识。  
这里需要注意的是这两个元素

```
<otherProperty name="hbm2ddl" value="create-update" />
```



```
<otherProperty name="format_sql" value="true" />
```

在xdoclet1.0.4以后的版本中hbm2ddl需要以额外的方式来设置,之前我按照1.0.4版本中的方式去设置,此属性死活不会出现在hibernate.cfg.xml中,最后得知这是xdoclet的一个bug

详见: <http://jira.codehaus.org/browse/MOJO-466>

打开mysql,创建一个名为xdoclet的数据库

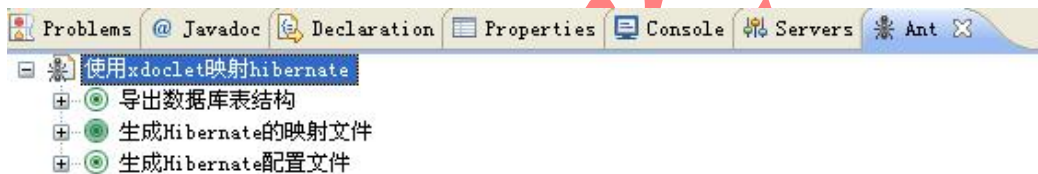
在eclipse中运行ant

点window->show view->ant



添加我们的ant脚本到eclipse的ant视图中

点击 , 选择build.xml文件



在导出数据库表结构上点击run as ant

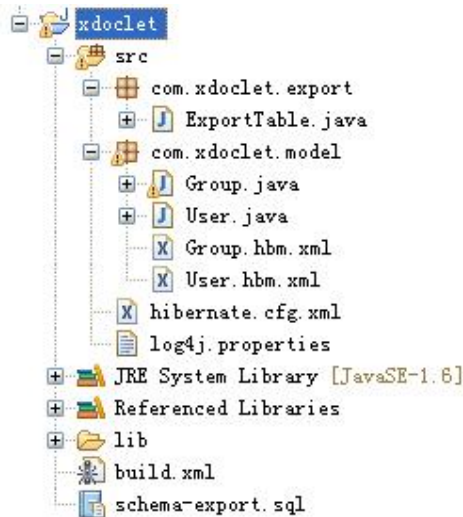
此时控制台输出:

```
Buildfile: E:\JAR\jbpm\jbpm-4.3\workspace\xdoclet\build.xml
生成Hibernate的映射文件:
[hibernatedoclet] (XDocletMain.start          47 ) Running
<hibernate/>
[hibernatedoclet] Generating mapping file for
com.xdoclet.model.Group.
[hibernatedoclet]    com.xdoclet.model.Group
[hibernatedoclet] Generating mapping file for
com.xdoclet.model.User.
[hibernatedoclet]    com.xdoclet.model.User
生成Hibernate配置文件:
[hibernatedoclet] addOtherProperty(): name=null, null
[hibernatedoclet] addOtherProperty(): name=null, null
```

```
[hibernatedoclet] (XDocletMain.start 47 ) Running
<hibernatecfg/>
[hibernatedoclet] Generating hibernate.cfg.xml configuration file
导出数据库表结构:
[schemaexport] (cfg.Environment 500 ) Hibernate
3.2.0
[schemaexport] (cfg.Environment 533 )
hibernate.properties not found
[schemaexport] (cfg.Environment 667 ) Bytecode
provider name : cglib
[schemaexport] (cfg.Environment 584 ) using JDK 1.4
java.sql.Timestamp handling
[schemaexport] (cfg.Configuration 274 ) Reading
mappings from file:
E:\JAR\jbpm\jbpm-4.3\workspace\xdoclet\src\com\xdoclet\model\Group.hbm.xml
[schemaexport] (cfg.HbmBinder 300 ) Mapping class:
com.xdoclet.model.Group -> t_group
[schemaexport] (cfg.Configuration 274 ) Reading
mappings from file:
E:\JAR\jbpm\jbpm-4.3\workspace\xdoclet\src\com\xdoclet\model\User.hbm.xml
[schemaexport] (cfg.HbmBinder 300 ) Mapping class:
com.xdoclet.model.User -> t_user
[schemaexport] (dialect.Dialect 141 ) Using
dialect: org.hibernate.dialect.MySQL5Dialect
[schemaexport] (cfg.HbmBinder 2375) Mapping
collection: com.xdoclet.model.Group.userSets -> t_user
[schemaexport] (hbm2ddl.SchemaExport 154 ) Running
hbm2ddl schema export
[schemaexport] (hbm2ddl.SchemaExport 174 ) writing
generated schema to file:
E:\JAR\jbpm\jbpm-4.3\workspace\xdoclet\schema-export.sql
[schemaexport]
[schemaexport] alter table t_user
[schemaexport] drop
[schemaexport] foreign key FKCB63CCB6CEAB0634;
[schemaexport]
[schemaexport] drop table if exists t_group;
[schemaexport]
[schemaexport] drop table if exists t_user;
[schemaexport]
[schemaexport] create table t_group (
[schemaexport] groupId varchar(255) not null,
```

```
[schemaexport]      groupName varchar(255),
[schemaexport]      primary key (groupId)
[schemaexport]      );
[schemaexport]
[schemaexport]      create table t_user (
[schemaexport]      userId varchar(255) not null,
[schemaexport]      userName varchar(255),
[schemaexport]      groupId varchar(255),
[schemaexport]      primary key (userId)
[schemaexport]      );
[schemaexport]
[schemaexport]      alter table t_user
[schemaexport]      add index FKCB63CCB6CEAB0634 (groupId),
[schemaexport]      add constraint FKCB63CCB6CEAB0634
[schemaexport]      foreign key (groupId)
[schemaexport]      references t_group (groupId);
[schemaexport] (hbm2ddl.SchemaExport 196 ) schema
export complete
BUILD SUCCESSFUL
Total time: 1 second
```

再观察项目目录结构



是不是奇迹出现了?不错, 我们做了一大堆工作要的就是这个结果

Group.hbm.xml文件内容

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping
>
    <class
```

```
name="com.xdoclet.model.Group"
table="t_group"
>
<id
  name="groupId"
  column="groupId"
  type="java.lang.String"
>
  <generator class="assigned">
    <!--
      To add non XDoclet generator parameters, create a file named
      hibernate-generator-params-Group.xml
      containing the additional parameters and place it in your merge
      dir.
    -->
  </generator>
</id>
<property
  name="groupName"
  type="java.lang.String"
  update="true"
  insert="true"
  column="groupName"
/>
<set
  name="userSets"
  lazy="false"
  inverse="true"
  cascade="none"
  sort="unsorted"
>
  <key
    column="groupId"
  >
  </key>
  <one-to-many
    class="com.xdoclet.model.User"
  />
</set>
<!--
  To add non XDoclet property mappings, create a file named
  hibernate-properties-Group.xml
  containing the additional properties and place it in your merge dir.
-->
```

```
</class>
</hibernate-mapping>
```

User.hbm.xml文件内容

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping
>
    <class
        name="com.xdoclet.model.User"
        table="t_user"
    >
        <id
            name="userId"
            column="userId"
            type="java.lang.String"
        >
            <generator class="assigned">
                <!--
                    To add non XDoclet generator parameters, create a file
named
                    hibernate-generator-params-User.xml
                    containing the additional parameters and place it in
your merge dir.
                -->
            </generator>
        </id>
        <property
            name="userName"
            type="java.lang.String"
            update="true"
            insert="true"
            column="userName"
        />
        <many-to-one
            name="group"
            class="com.xdoclet.model.Group"
            cascade="all"
            outer-join="auto"
            update="true"
            insert="true"
            column="groupId"
        />
    </class>
</hibernate-mapping>
```

```
<!--
    To add non XDoclet property mappings, create a file named
        hibernate-properties-User.xml
    containing the additional properties and place it in your
merge dir.
-->

</class>
</hibernate-mapping>
```

生成的hibernate.cfg.xml文件内容

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate
Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd"
">
<!-- Generated file - Do not edit! -->
<hibernate-configuration>
    <!-- a SessionFactory instance listed as /jndi/name -->
    <session-factory>
        <!-- properties -->
        <property
name="dialect">org.hibernate.dialect.MySQL5Dialect</property>
        <property name="show_sql">true</property>
        <property name="use_outer_join">false</property>
            <property name="connection.username">root</property>
            <property name="connection.password">root</property>
            <property
name="connection.driver_class">com.mysql.jdbc.Driver</property>
            <property
name="connection.url">jdbc:mysql://localhost:3306/xdoclet</proper
ty>
            <property name="hbm2ddl">create-update</property>
            <property name="format_sql">true</property>
            <!-- mapping files -->
            <mapping resource="com/xdoclet/model/Group.hbm.xml"/>
            <mapping resource="com/xdoclet/model/User.hbm.xml"/>
        </session-factory>
    </hibernate-configuration>
```

是不是跟预期的效果一样呢?

更为激动的是sql脚本

```
alter table t_user
drop
foreign key FKCB63CCB6CEAB0634;
drop table if exists t_group;
drop table if exists t_user;
```

```
create table t_group (  
    groupId varchar(255) not null,  
    groupName varchar(255),  
    primary key (groupId)  
);  
  
create table t_user (  
    userId varchar(255) not null,  
    userName varchar(255),  
    groupId varchar(255),  
    primary key (userId)  
);  
  
alter table t_user  
    add index FKCB63CCB6CEAB0634 (groupId),  
    add constraint FKCB63CCB6CEAB0634  
    foreign key (groupId)  
    references t_group (groupId);
```

紧接着就来介绍ExportTable.java文件

```
package com.xdoclet.export;  
import org.hibernate.cfg.Configuration;  
import org.hibernate.tool.hbm2ddl.SchemaExport;  
  
public class ExportTable {  
  
    public static void main(String[] args) {  
        Configuration configuration=new  
Configuration().configure("hibernate.cfg.xml");  
        SchemaExport export=new SchemaExport(configuration);  
        export.create(true, true);  
    }  
}
```

很简单,一看就明白了,赶紧run一把吧!

查看生成的数据库表结构



The screenshot shows a Windows command prompt window with the title bar '选定 C:\WINDOWS\system32\cmd.exe - mysql -uroot -proot'. The command prompt shows the following sequence of commands and output:

```
mysql> use xdoclet  
Database changed  
mysql> show tables;  
Empty set <0.00 sec>  
  
mysql> show tables;  
+-----+  
! Tables_in_xdoclet !  
+-----+  
! t_group            !  
! t_user             !  
+-----+  
2 rows in set <0.00 sec>
```

T\_user

```
C:\WINDOWS\system32\cmd.exe - mysql -uroot -proot

mysql> desc t_user;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| userId     | varchar(255)  | NO   | PRI | NULL    |       |
| userName   | varchar(255)  | YES  |     | NULL    |       |
| groupId    | varchar(255)  | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> _
```

T\_group

```
C:\WINDOWS\system32\cmd.exe - mysql -uroot -proot

mysql> desc T_group;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| groupId    | varchar(255)  | NO   | PRI | NULL    |       |
| groupName  | varchar(255)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

哈哈，终于写完了。

现在你可以任意删除\*.hbm.xml和hibernate.cfg.xml文件,只需重新运行ant就会再次生成\*.hbm.xml和hibernate.cfg.xml文件，是不是很方便？