**快速开发平台之---代码加速器**

目录

[一、 代码加速器的介绍 2](#_Toc331508890)

[二、 使用说明 2](#_Toc331508891)

[1. 在项目中引入代码加速器 2](#_Toc331508892)

[2. 生成代码目录结构及设置说明 2](#_Toc331508893)

[1) 目录结构及文件说明 2](#_Toc331508894)

[2) 生成代码设置步骤 4](#_Toc331508895)

[3) 生成代码说明 6](#_Toc331508896)

[三、 关于MySql版本的代码变更说明 9](#_Toc331508897)

[1. 执行metadata/db/mysql下的建库脚本 9](#_Toc331508898)

[2. 修改resources/conf/jdbc.properties里的内容 9](#_Toc331508899)

[3. Hibernate的映射实体转化MySql的映射类型 10](#_Toc331508900)

[4. 代码生成器的修改说明 10](#_Toc331508901)

[四、 配置struts.xml 11](#_Toc331508902)

[五、 配置menu-oa.xml 11](#_Toc331508903)

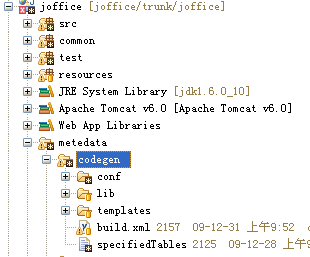
# 代码加速器的介绍

代码加速器是基于Velocity模板，Ant执行脚本，结合项目的开发，可以大大减少开发人员的工作量，把重复的程序配置、通用的功能（ＣＲＵＤ）管理等代码一起生成出来。开发用户仅需要设计好相关的表，配置好生成代码的参数，执行Ant的脚本命令则可。基于此，开发人员可以从重复繁杂的编码工作中解放出来，专注于程序的业务实现。目前代码加速器的模板是基于J.Office的开发平台上的，开发用户也可以根据自己的需要，修改模板，添加模板或删除模板等。

# 使用说明

## 在项目中引入代码加速器

把代码加速器的全部代码放置在开发目录下，如可以放置在metadata目录下，如下图所示：

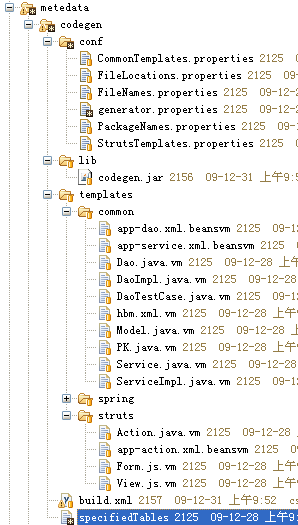


设置好目录后，需要指定代码生成的根目录，即我们需要把代码生成到该目录的上两级目录，在这里，我们需要进入代码加速器的配置。

## 生成代码目录结构、步骤设置说明

### 目录结构及文件说明

代码加速器的目录结构如下所示



#### conf目录

代码加速器的相关配置目录

#### lib目录

为代码加速器的依赖包

#### templates目录

为代码生成的模板目录，其下的common目录为通用的应用程序模板，在开发JEE程序时，我们都会有相应的实体类，服务类，数据访问类等，而视图层可以采用多种技术，如Spring MVC 、Struts，所以其下的另外两个目录为SpringMVC、Struts视图的模板。

#### build.xml

为代码生成器的脚本命令配置

#### specifiedTables

用于指定生成代码的表

### 生成代码设置步骤

#### 设置代码生成的目录

修改conf/generator.properties下的javaOutputLocation参数，使其设置为以下值：

javaOutputLocation=../../

在此也可以写成开发环境的项目的绝对路径，如：

javaOutputLocation= D:/workspace/joffice/

#### 设置数据库连接

修改conf/generator.properties下的以下参数，并设置正确，用户名、密码及Schema如：

### Database Settings ###

# [All Databases]

dbUserid=joffice

dbPasswd= joffice

DbSchema=orcl

找到你要连接的数据库，例如Oracle连接：

jdbcDriver=oracle.jdbc.OracleDriver

dbUrl=jdbc:oracle:thin:@localhost:1521:ORCL

useCaseSensitiveNames=False

【注】在Oracle的环境下，需要特别注意，由于在oracle下，其默认字段是不区分大小写的，因而这对生成的代码并不好，如某实体的属性字段均为大写会给代码带来一些问题。所以可以建多一个Oracle数据库用于生成代码，并设置其区分大小写字段。这样还不与其维护多一个mysql的库，用于生成代码更为方便些，两个库的不同脚本可以通过powerdesign工具可以比较方便进行转化。

#### 设置实体所在包

修改conf/PackageNames.properties，加上类似以下代码，告诉生成器把代码生成至哪一个目录包下(左为表名，右为包名): 例如表名 msg\_info 和msg\_type在 system包下，如下：

msg\_info=system

msg\_type=system

duty=personal

errands\_register=personal

#### 指定生成代码对应的表

在specifiedTables指定生成的表名，如以下则代表同时生成这两个表

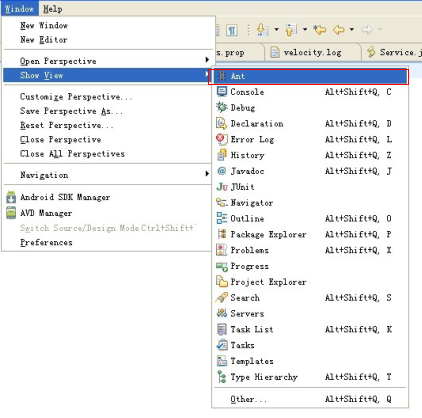
#注意：Oracle一定要大写

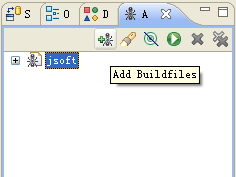
msg\_info

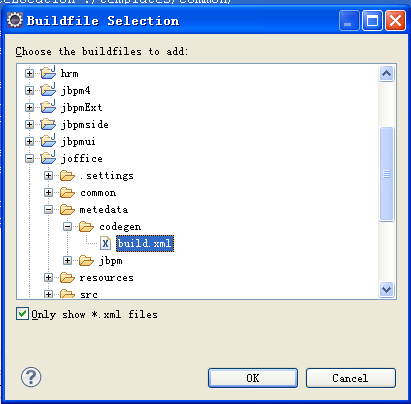
msg\_type

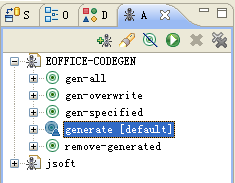
#### 执行Ant的命令

在eclipse中的Ant视图中，把codegen/build.xml加进去，执行其缺省的任务（generate）







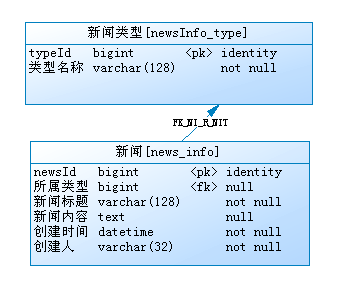


【说明】

以上步骤１、２仅需要设置一次，其后产生其他表时，则直接设置步骤３、４则可。生成的文件及路径由conf/ CommonTemplates.properties及conf/FileLoaction.properties两个文件来设置指定的，开发用户若需要生成其他文件，也可以在这两个文件中进行指定。

### 生成代码说明

设计表



如我们生成管理news\_info表时，其产生的代码包括：

1. **实体类**：NewsInfo.java
2. **映射文件**：News Info.hbm.xml
3. **Dao接口及实现类**：

NewsInfoDao.java

NewsInfoDaoImpl.java

1. **Service接口及实现类**：News InfoService.java

NewsInfoServiceImpl.java

1. **测试类**：

NewsInfoDaoTestCase.java

1. **配置文件**：

app-service.xml

app-dao.xml

app-action.xml

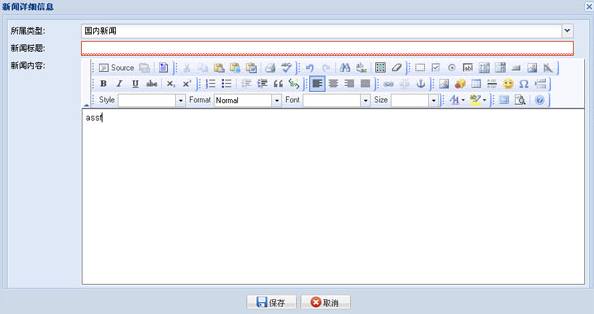
1. **Ext js**

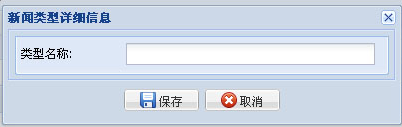
NewsInfoView.js

NewsInfoForm.js

生成的最终功能界面如下所示：





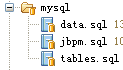




# 关于MySql版本的代码变更说明

本代码结构原为Oracle版本，若变更为MySQL版本，需要进行以下步骤进行修改

## 执行metadata/db/mysql下的建库脚本



## 修改resources/conf/jdbc.properties里的内容

注意：MySql的数据库版本，若为5.1（不包括5.1）之前的版本，

其hibernate.dialect设置为：

hibernate.dialect=org.hibernate.dialect.MySQLInnoDBDialect

5.1之后的版本，则应该设置为：

hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect

如下所示:

jdbc.driverClassName=com.mysql.jdbc.Driver

jdbc.url=jdbc:mysql://localhost:3306/oa?useUnicode=true&amp;characterEncoding=utf-8

hibernate.dialect=org.hibernate.dialect.MySQLInnoDBDialect

#org.hibernate.dialect.MySQL5InnoDBDialect for mysql 5.5

jdbc.username=root

jdbc.password=root

## Hibernate的映射实体转化MySql的映射类型

在eclipse的ant视图中，打开metadata\setup\hbm\_converter.xml，执行其toNormal任务，表示把Hibernate的映射实体转化MySql或SqlServer的映射类型。

## 代码生成器的修改说明

修改metadata\codegen\generator.properties下的mysql连接，如把以下：

jdbcDriver=oracle.jdbc.OracleDriver

dbUrl=jdbc:oracle:thin:@localhost:1521:ORCL

useCaseSensitiveNames=False

换成

jdbcDriver=com.mysql.jdbc.Driver

dbUrl=jdbc:mysql://localhost:3306/joffice2

useCaseSensitiveNames=True

同时修改templates\common\hbm.xml.vm的58行，生成主键的代码换成下面代码

<generator class="native"/>

##<generator class="sequence">

## <param name="sequence">S\_${sqlTable.table}</param>

##</generator>

# 配置struts.xml

如加入到报名system下

<package name=*"system"* extends=*"struts-default"* namespace=*"/system"*>

<action name=*"\*AppAreaTest"* class=*"AppAreaTestAction"* method=*"{1}"*>

<result>${successResultValue}</result>

</action>

<action name=*"\*AppUserTest"* class=*"AppUserTestAction"* method=*"{1}"*>

<result>${successResultValue}</result>

</action>

</ package>

# 配置menu-oa.xml

如加入菜单到栏目我的桌面下

<Items id=*"MyDesktop"* text=*"我的桌面"* iconCls=*"menu-desktop"* isPublic=*"false"*>

<Item id=*"AppAreaTestView"* text=*"区域管理测试"* iconCls=*"menu-attachment"*/>

<Item id=*"AppUserTestView"* text=*"用户管理测试"* iconCls=*"menu-attachment"*/>

</Items>