# EOS批处理框架配置及使用说明

本文档指导在EOS6.0基础上，集成批处理框架的步骤。

## 一、安装EOS6.0

1、安装Platform 6.0 ，选择report和BPS模块(注意在Win7下需更换jdk)

2、在安装好的文件目录的下的IDE/eclipse/plugins中找到org.eclipse.swt开头的两个Jar包文件，应该是:

org.eclipse.swt.win32.win32.x86\_3.2.1.v3235.jar

org.eclipse.swt\_3.2.1.v3235e.jar

3、备份这两个jar文件，将附件中的jar文件分别替换这两个jar。

## 二、打补丁

安装EOS6.0成功之后，按以下步骤打补丁：

1、拷贝运行期spring相关JAR

|  |  |
| --- | --- |
| 源 | 目标 |
| %CVS\_ROOT%/develop/server/build/batch/eos-spring.jar | %EOS\_HOME%\apache-tomcat-5.5.20\webapps\eos-default\WEB-INF\lib |
| %CVS\_ROOT%/develop/server/build/batch/spring-\*.jar、com.springsource.org.codehaus.jettison-1.0.0.jar、aopalliance-1.0.jar | %EOS\_HOME%\apache-tomcat-5.5.20\common\lib |

2、IDE插件

|  |  |
| --- | --- |
| 源 | 目标 |
| %CVS\_ROOT%\develop\ide\build\batch\\*.jar | %EOS\_HOME%\ide\eostools\eclipse\plugins |

3、拷贝开发期jar依赖

|  |  |
| --- | --- |
| 源 | 目标 |
| %CVS\_ROOT%/spring-batch-core-2.1.9.RELEASE.jar、spring-batch-infrastructure-2.1.9.RELEASE.jar | %EOS\_HOME%\ide\eostools\eclipse\plugins\com.primeton.studio.server.library\_6.0.0\server\3rd\spring |
| %CVS\_ROOT%/develop/server/build/batch/eos-spring.jar | %EOS\_HOME%\ide\eostools\eclipse\plugins\com.primeton.studio.server.library\_6.0.0\server\eos |

4、系统配置

|  |  |
| --- | --- |
| 源 | 目标 |
| %CVS\_ROOT%\develop\server\src\batch\config\batch-system-config.xml | %EOS\_HOME%\eosserver\working\eos-default\config |

## 三、batch-admin的部署（可选）

batch-admin是批处理框架的监控平台，可以对批处理框架进行管控。

部署步骤：

1、解压%CVS\_ROOT%\develop\server\build\batch\spring-batch-admin.war至tomcat

2、修改数据库连接配置：

文件位置：spring-batch-admin\WEB-INF\classes\batch-oracle.properties

一般情况下，和批处理框架的数据库JOB仓库共用同一数据源。

3、增加-D参数，指定使用的数据库类型

-DENVIRONMENT=oracle

4、修改batch-system-config.xml使用基于数据库的JOB仓库

文件位置：%EOS\_HOME%\eosserver\working\eos-default\config\batch-system-config.xml

打开org.springframework.batch.core.repository.support.JobRepositoryFactoryBean的注释，以使用基于数据库的JOB仓库，运行期JOB相关数据都会持久化到数据库。

## 四、批处理开发过程

1、定义批处理流程

在构件包->配置上右键，创建批处理

示例如下：

|  |
| --- |
| <job id="custinfo\_batch\_concurrent">  <step id="partitionStep">  <partition step="csvReadWriter" partitioner="partitionerDB">  <handler grid-size="5" task-executor="taskExecutor"/>  </partition>  </step>  </job>  <step id="csvReadWriter">  <tasklet transaction-manager="transactionManager" >  <chunk reader="jdbcItemPageReader" writer="messageWriter" commit-interval="2000" chunk-completion-policy="" />  </tasklet>  </step>  <beans:bean id="messageWriter" class="org.springframework.batch.item.file.FlatFileItemWriter" scope="step">  <beans:property name="resource" value="file:#{stepExecutionContext[\_fileName]}"/>  <beans:property name="lineAggregator">  <beans:bean class="com.primeton.poc.nbyh.batch.MessageExtractorLineAggregator"></beans:bean>  </beans:property>  </beans:bean>  <beans:bean id="taskExecutor" class="org.springframework.scheduling.concurrent.ThreadPoolTaskExecutor">  <beans:property name="corePoolSize" value="5"/>  <beans:property name="maxPoolSize" value="5"/>  </beans:bean>  <beans:bean id="partitionerDB" class="com.primeton.spring.batch.EOSBatchPartitionForDB" scope="step">  <beans:property name="table" value="CUSTINFO"/>  <beans:property name="column" value="KHYXID"/>  <beans:property name="dataSource" ref="dataSource"/>  <beans:property name="outputFileName" value="#{jobParameters['WAR\_PATH']}/output.csv"/>  </beans:bean>  <beans:bean id="jdbcItemPageReader" class="org.springframework.batch.item.database.JdbcPagingItemReader" scope="step">  <beans:property name="dataSource" ref="dataSource"/>  <beans:property name="queryProvider">  <beans:bean class="org.springframework.batch.item.database.support.SqlPagingQueryProviderFactoryBean">  <beans:property name="dataSource" ref="dataSource"/>  <beans:property name="selectClause" value="select CUSTINFO.\*"/>  <beans:property name="fromClause" value="from CUSTINFO"/>  <beans:property name="whereClause" value="where KHYXID between #{stepExecutionContext[\_minRecord]} and #{stepExecutionContext[\_maxRecord]} and #{jobParameters['WHERE\_CLAUSE']}"/>  <beans:property name="sortKey" value="KHYXID"/>  </beans:bean>  </beans:property>  <beans:property name="pageSize" value="10"/>  <beans:property name="rowMapper">  <beans:bean id="userDbMapper" class="com.primeton.spring.batch.SDODbMapper">  <beans:property name="uri" value="com.primeton.poc.nbyh.custInfo" />  <beans:property name="typeName" value="Custinfo" />  </beans:bean>  </beans:property>  </beans:bean> |

## 五、API

|  |  |
| --- | --- |
| 类 | 说明 |
| com.primeton.spring.batch. EOSBatchTaskSchedule | 批处理任务的执行调度器 |
| com.primeton.spring.batch. EOSBatchPartitionForDB | 基于数据库的分片器，用于并行执行任务时针对数据库表的逻辑划分 |
| com.primeton.spring.batch. SDODbMapper | 基于SDO的列绑定 |