

# 关于 TestToolTaskProject

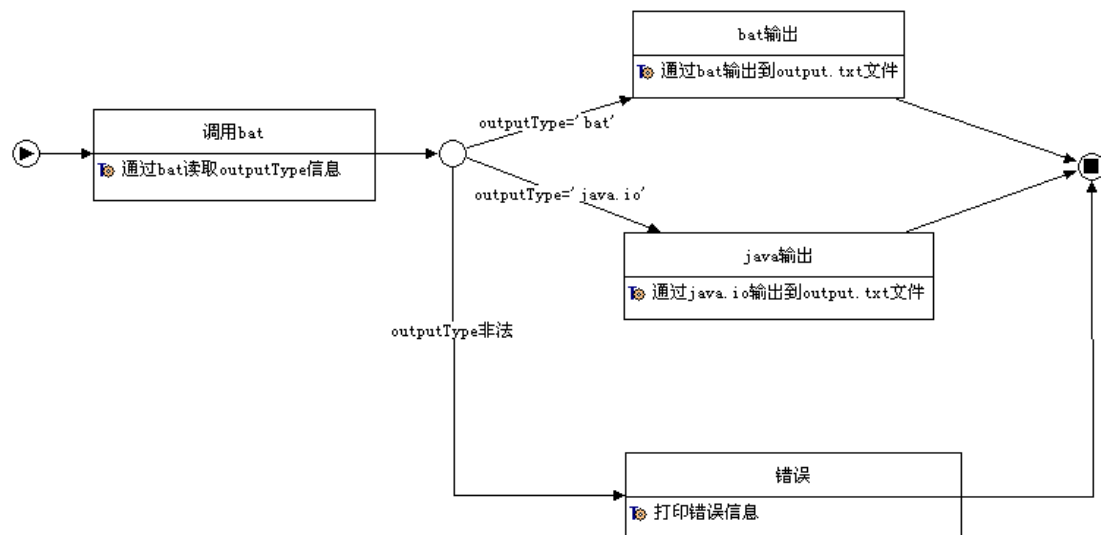
1、该 Project 的目的.....	1
2、业务描述.....	1
3、如何运行该 Example.....	2
4、代码分析.....	2
4.1 project 目录结构.....	2
4.2 TestToolTaskMain 启动流程实例.....	3
4.3 流程定义文件的加载.....	3
4.4 流程实例的存储.....	4

## 1、该 Project 的目的

- 1) 这个 project 展示了如何使用 ToolTask,
- 2) 同时展示了如何将 Fire workflow 嵌入 J2SE 系统中。

## 2、业务描述

该业务的流程图如下



系统首先通过批处理程序 `getOutputType.bat` 读取 `outputType.txt` 文件中的 `outputType` 信息。`outputType` 的合法取值有两个，一个是“bat”，另一个是“java.io”。

当 `outputType` 等于“bat”时，系统调用另一个 `output.bat` 批处理程序，往文件 `output.txt` 中输出“Hello World”信息。

当 outputType 等于 “java.io” 时，系统调用 java 的 IO 往 output.txt 输出 “Hello World” 信息。

当 ouputType 非法时，打印错误信息。

## 3、如何运行该 Example

该程序的主类是 org.fireflow.example.testtooltask. TestToolTaskMain。

运行后，系统首先启动一个 hsqldb 数据库，用于存储流程实例信息。然后启动并运行流程。运行结束后关闭 hsqldb 数据库。

流程实例运行时打印出相关的提示信息如下图

```
2009-7-9 15:40:09 org.hibernate.impl.SessionFactoryImpl <init>
信息: building session factory
2009-7-9 15:40:09 org.hibernate.impl.SessionFactoryObjectFactory addInstance
信息: Not binding factory to JNDI, no JNDI name configured
2009-7-9 15:40:09 org.springframework.orm.hibernate3.HibernateTransactionManager aft
信息: Using DataSource [org.apache.commons.dbcp.BasicDataSource@18f1be9] of Hibernate
=====启动流程实例，并执行.....
This is GetOutputTypeAppHandler::模拟调用bat文件，并取得返回值；根据返回值决定路由。
This is JavaioOutputAppHandler::模拟调用java程序
=====流程实例结束！
数据库服务已关闭！
[Server@a1807c]: Initiating shutdown sequence...
[Server@a1807c]: Shutdown sequence completed in 94 ms.
[Server@a1807c]: 2009-07-09 15:40:10.794 SHUTDOWN : System.exit() was not called
```

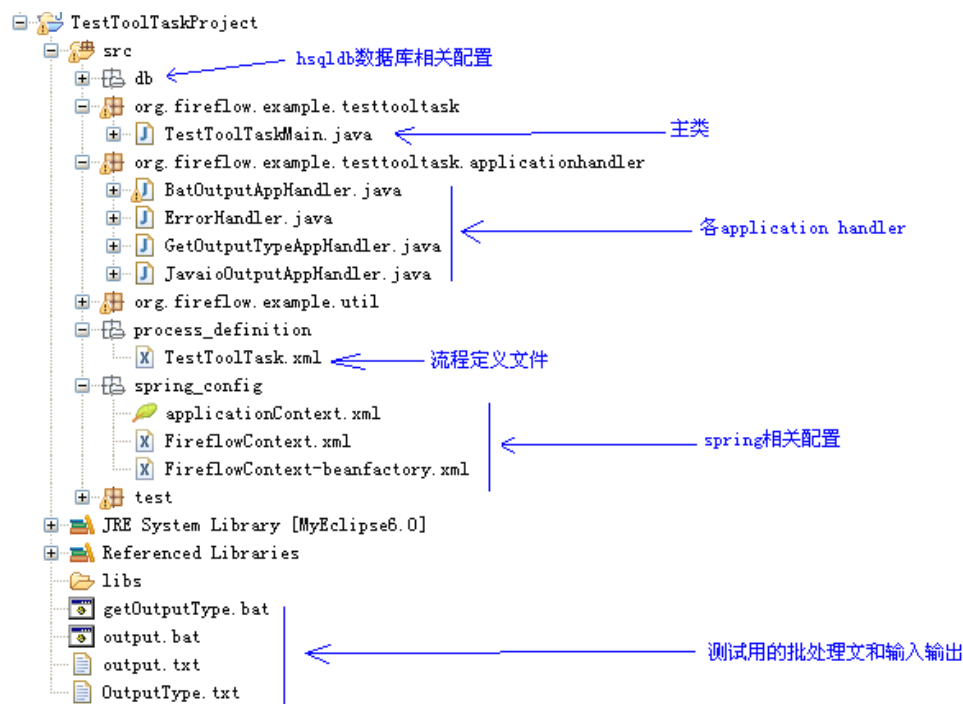
运行结束后，在项目根目录下产生 output.txt 文件，里面有相关的输出信息。

可以改变项目根目录下的 outputType.txt 文件中的 outputType 的值，再次运行程序，看看 output.txt 中的输出有什么不同。outputType 的合法取值有两个，一个是 “bat”，另一个是 “java.io”。

## 4、代码分析

### 4.1 project 目录结构

Project 目录结构如下图



## 4.2 TestToolTaskMain 启动流程实例

在 TestToolTaskMain 中启动流程实例相当简单，首先在 setUp()方法中取得 Fire workflow 的 runtimeContext 对象，然后，如下图红线所示 3 句话即可创建并运行流程实例。

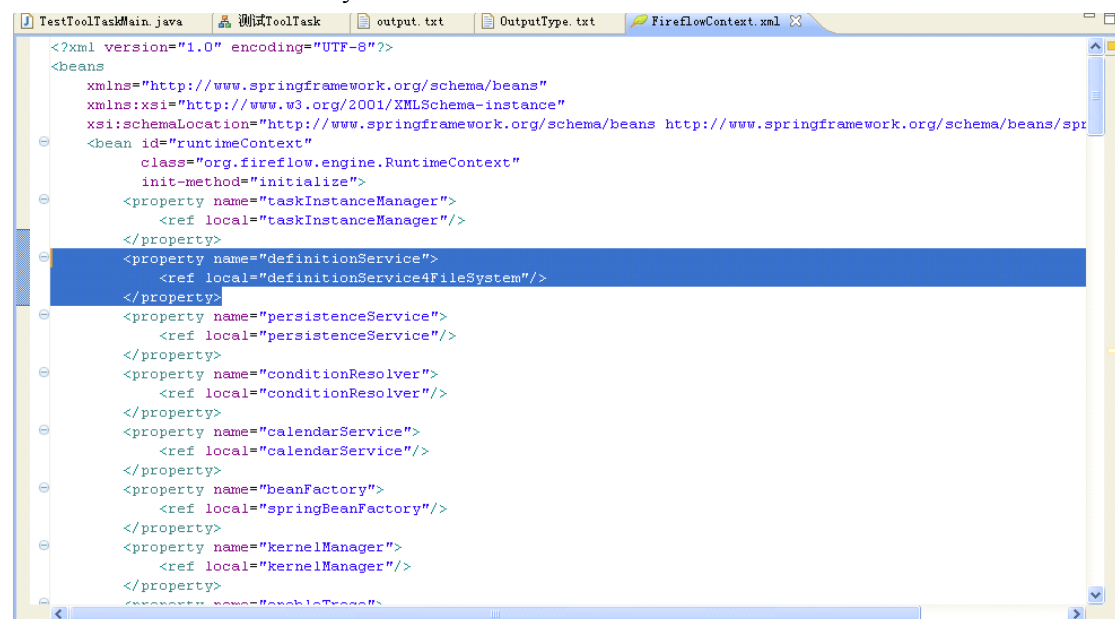
```
public static void main(String[] args) {  
    try {  
        // 启动数据库，初始化环境  
        setUp();  
  
        //启动流程实例，并执行该实例  
        System.out.println("=====启动流程实例，并执行.....");  
        transactionTemplate.execute(new TransactionCallback() {  
  
            public Object doInTransaction(TransactionStatus arg0) {  
                IWorkflowSession workflowSession = runtimeContext.getWorkflowSession();  
                try {  
                    IProcessInstance processInstance = workflowSession.createProcessInstance("TestToolTaskProce  
                    processInstance.run();  
                } catch (EngineException e) {  
                    // TODO Auto-generated catch block  
                    e.printStackTrace();  
                } catch (KernelException e) {  
                    // TODO Auto-generated catch block  
                    e.printStackTrace();  
                }  
  
                return null;  
            }  
        });  
        System.out.println("=====流程实例结束!");  
        //关闭数据库  
        tearDown();  
    } catch (Exception ex) {
```

## 4.3 流程定义文件的加载

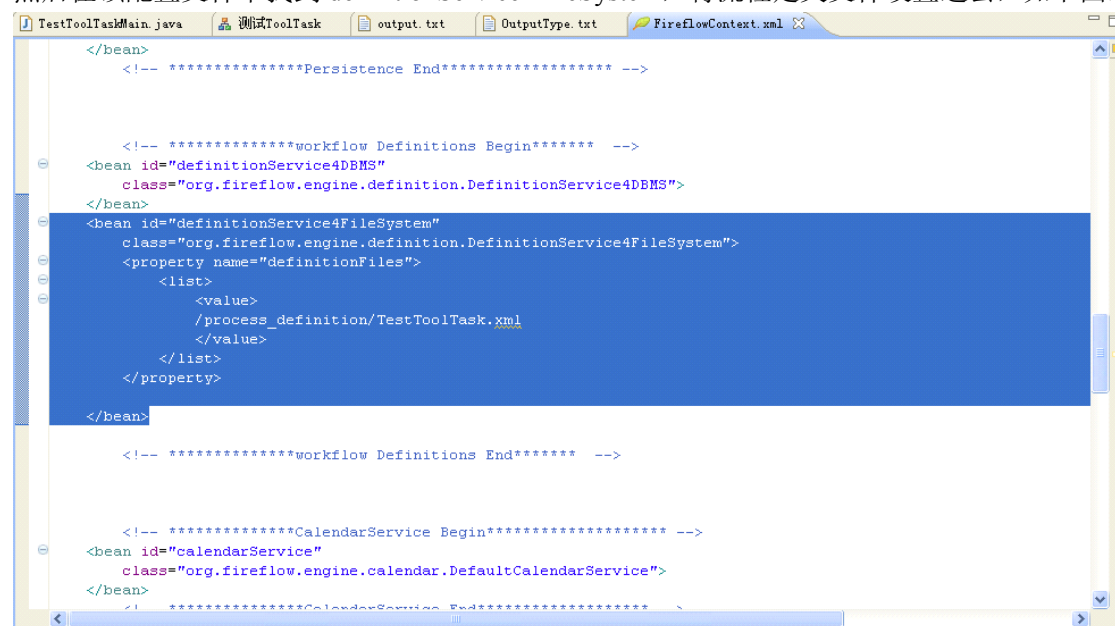
在本 Example 中流程定义文件没有保存在数据库中，而是保存在 project 的 bin 目录下，系

统通过加载 classpath 资源的方式读取流程定义文件。这种保存方法如下如下配置。

打开 FireflowContext.xml 配置文件，首先将 runtimeContext bean 的 definitionService 属性配置成 definitionService4FileSystem。如下图



然后在该配置文件中找到 definitionService4FileSystem，将流程定义文件设置进去，如下图。



## 4.4 流程实例的存储

在这个 Example 中，使用 hsqldb 内存数据库保存流程实例。其 hibernate 配置和普通数据库完全一致，唯一不同的是需要通过程序启动该数据库。启动和关闭数据库的代码在 org.fireflow.example.util.HsqldbManager