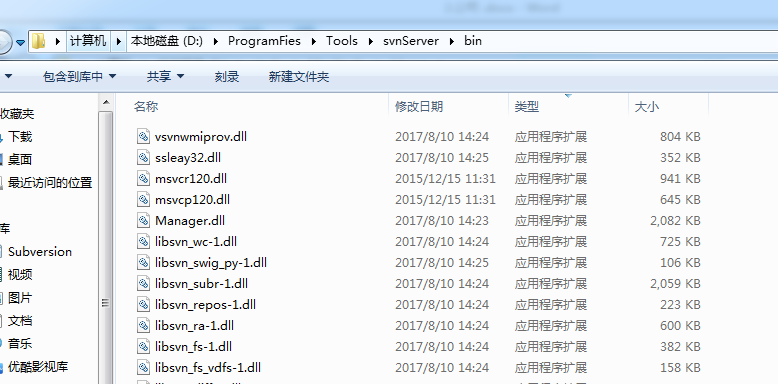
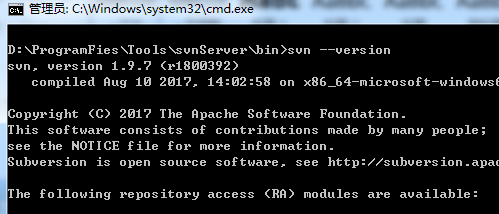
# 使用SVN命令

1、安装SVNserver服务器，再安装目录下打开bin的cmd窗口



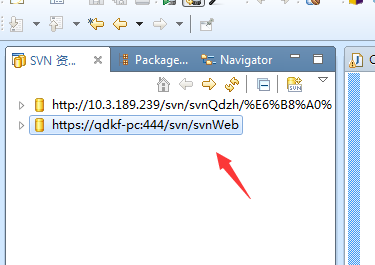


## 1、观察SVNserver是不是安装成功

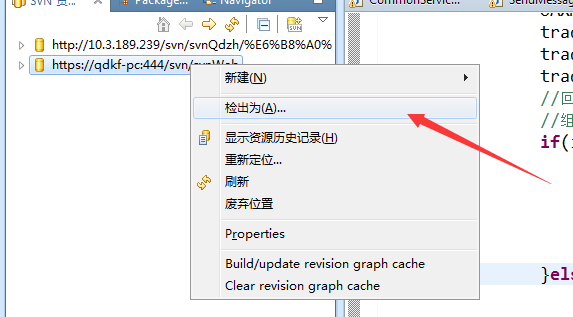
### svn –server

# 2、svnServer搭建项目，并且使用eclipse进行下载

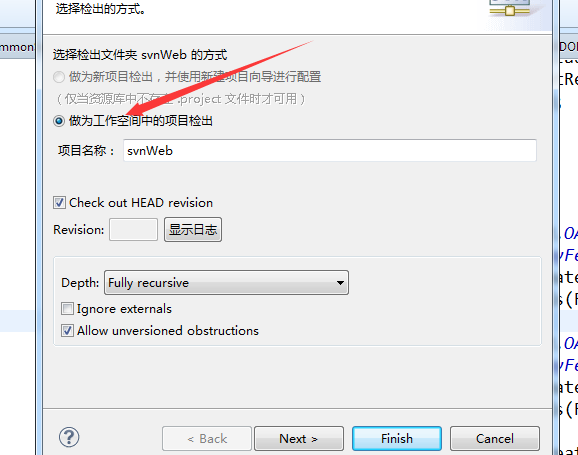
## 1、SVNserver 创建项目名svnWeb，然后我们在eclipse也创建一个项目svnWeb。将eclipse项目放到svnWeb下，然后删除eclipse项目，最后我们使用eclipse svn插件将SVNserver 项目连接



## 2、检出，下载SVNserver服务器代码，并放到工作空间



### 2、放到eclipse工作空间



### 3、检出成功

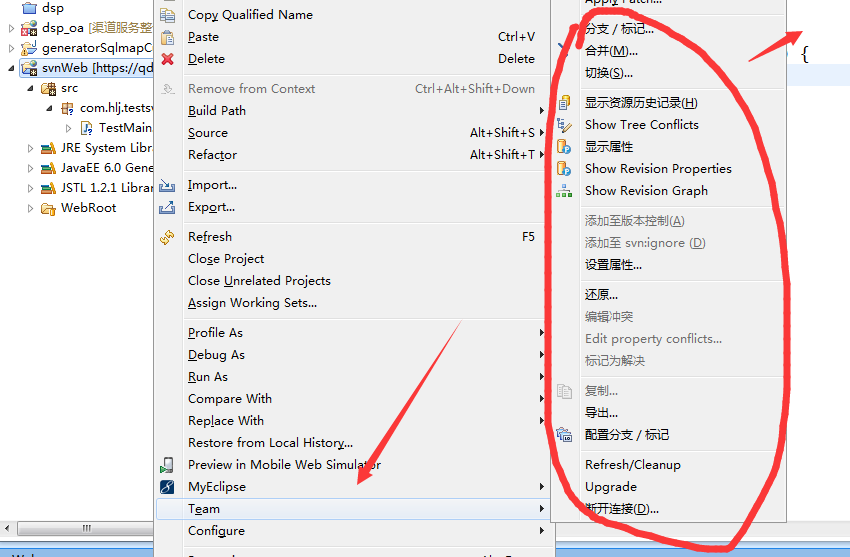


# 3、使用eclipse svn插件

### 1、测试，先创建一个包，和一个类 class



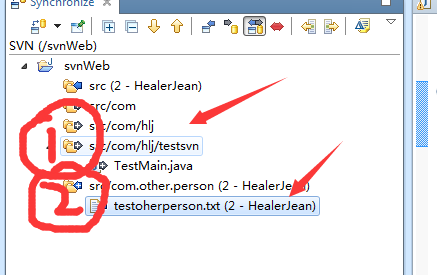
### 2、右键 观察右面的作用



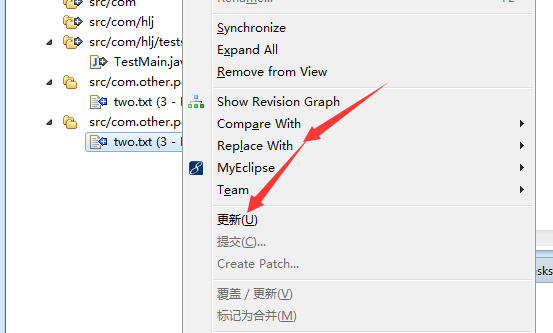
### 1、与资源库同步，这里会发现我们已经做过修改的，以及在资源库中别人提交的，我们还未拉下来的项目

#### 1、箭头 -》 是我们做过更改的项目

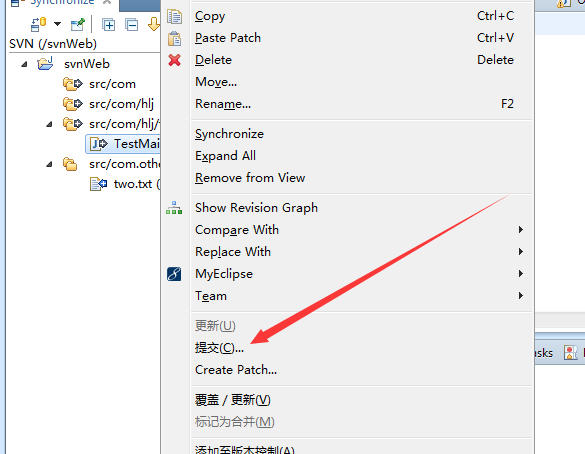
#### 2、箭头 《- 是别人提交到svn服务器，我们没有拉的



### 3、先拉去服务器端代码（我们已经添加了），右键-更新》 这样就会将服务器代码先拉到本地



### 4、提交本地到服务器 右键->提交

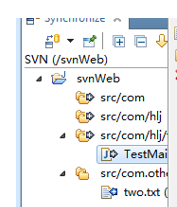


#### 4.2 填写注释，以及选择要提交的文件，ctrl+a 就会全选文件，然后选择其中一个的复选框即可全部选中

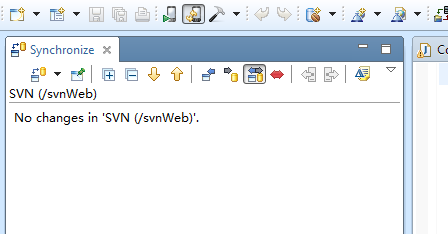


### 5、上面我们观察到，有层次文件夹会的显示出来，只要提交最下面的，中间的com hlj 等文件夹会自动帮我们提交

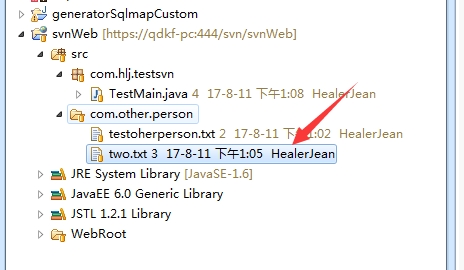
#### 1、未提交前



#### 2、本地提交后，现在

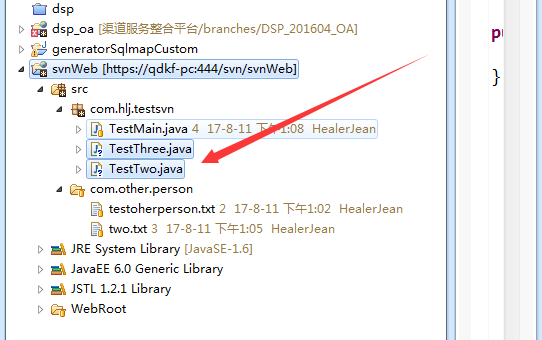


### 6、回到项目中会发现项目中提交的时间和人物(分支名称)

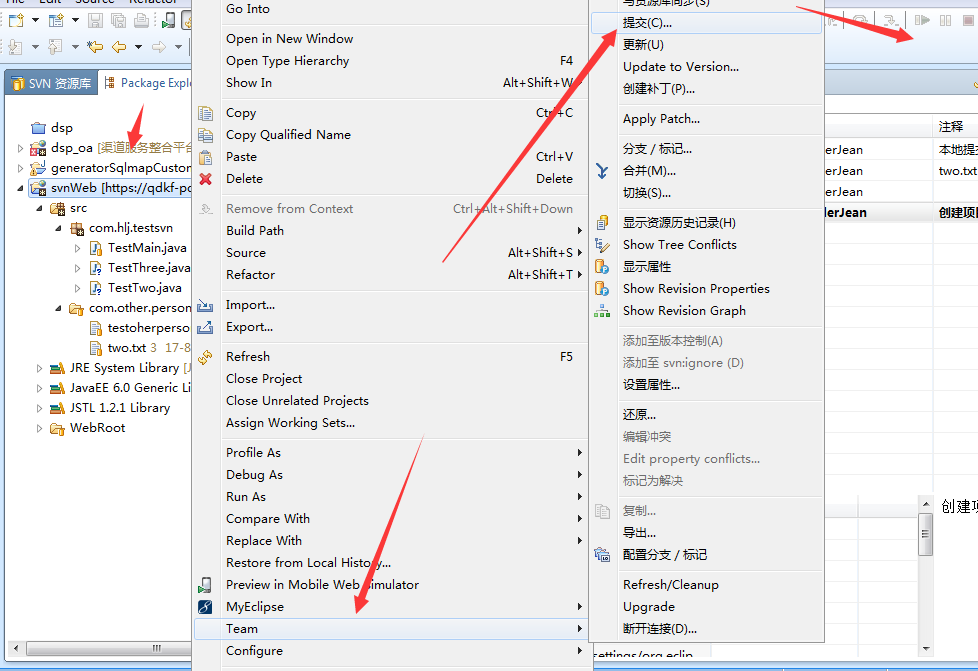


# 4、项目全部提交，直接项目的右键就能够全部提交

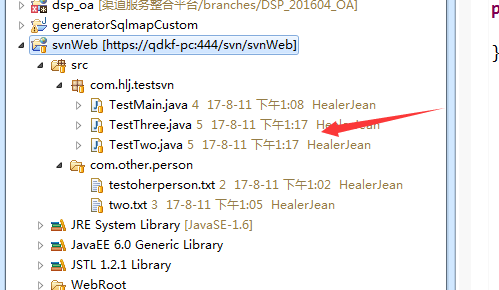
## 1、未提交之前，可以观察到后面是没有时间和分支名称的



## 2.右键->提交

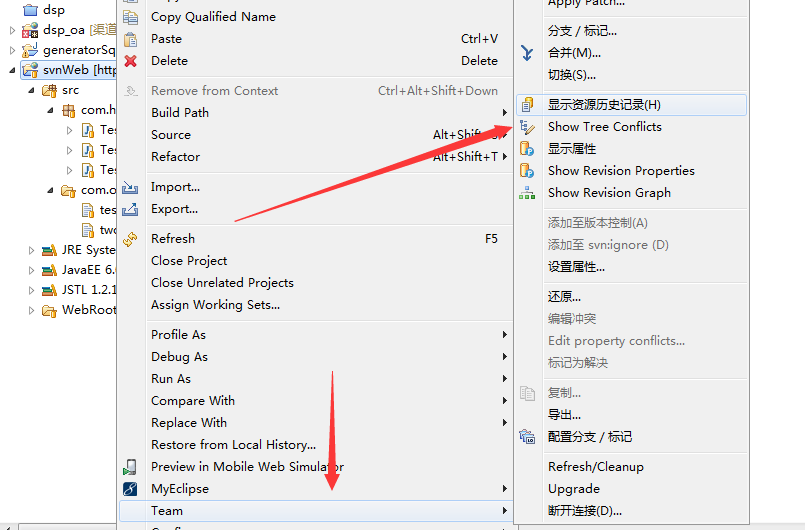


## 3、观察到成功了，因为有了时间和日期了

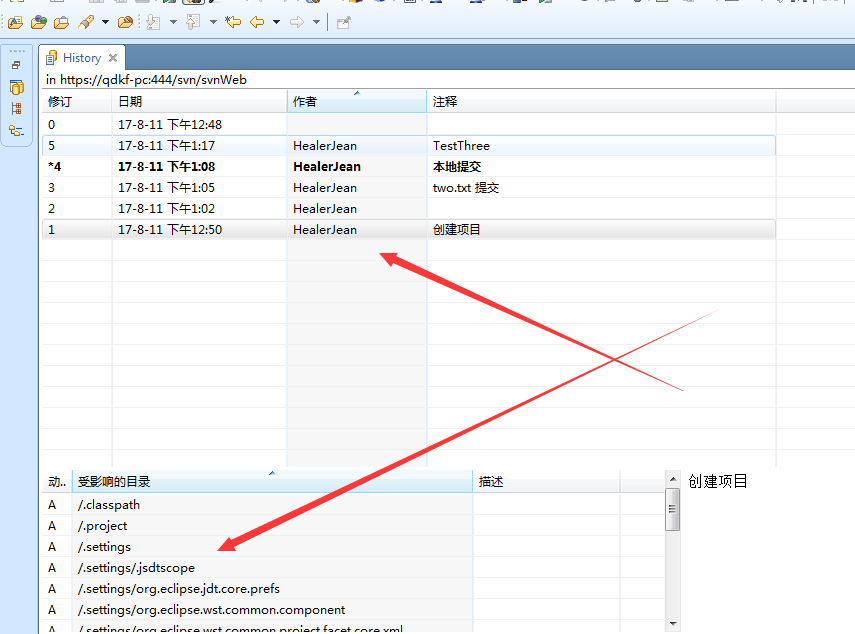


# 5、提交历史记录分析

## 1、打开方式

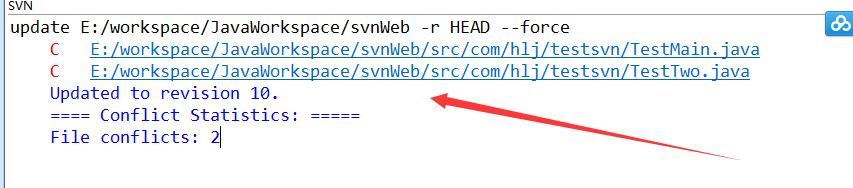


## 2.观察到提交者，时间，版本，提交文件内容

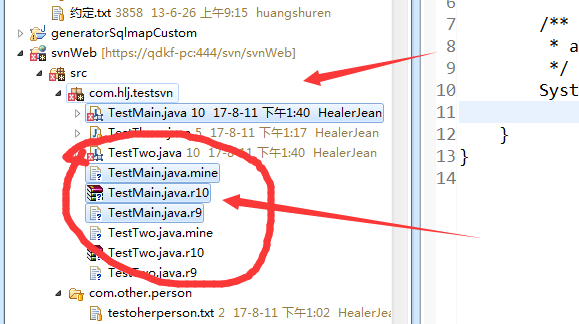


# 6.解决冲突

## 1、发生冲突征兆Eclipse控制台 以下显示2个冲突，文件也会给出来



## 2、关于发生冲突的文件会多增加3个文件



### 1、打开我们冲突的文件，上面的mine是我改的文件，也就是准备上传，结果上传晚了，被别人捷足先登。下面的就是别人修改的文件。

|  |
| --- |
| **public** **class** TestMain {  **public** **static** **void** main(String[] args) {  System.out.println("hello world");    <<<<<<< .mine  /\*\*  \* after commit  \*/  System.out.println("after commit ");  =======  /\*\*  \* before commit  \*/  System.out.println("before commit ");  >>>>>>> .r10    }  } |

### 2、mine文件，为我的文件代码，也就是后准备上传的文件，没有冲突，

|  |
| --- |
| public class TestMain {  public static void main(String[] args) {  System.out.println("hello world");    /\*\*  \* after commit  \*/  System.out.println("after commit ");    }  } |

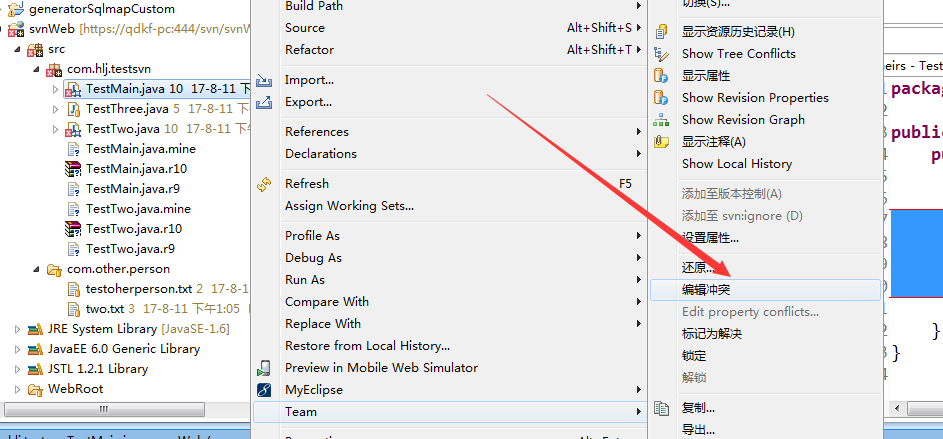
### 3、r9文件，为我和他都没有改动，之前的文件代码

|  |
| --- |
| public class TestMain {  public static void main(String[] args) {  System.out.println("hello world");      }  } |

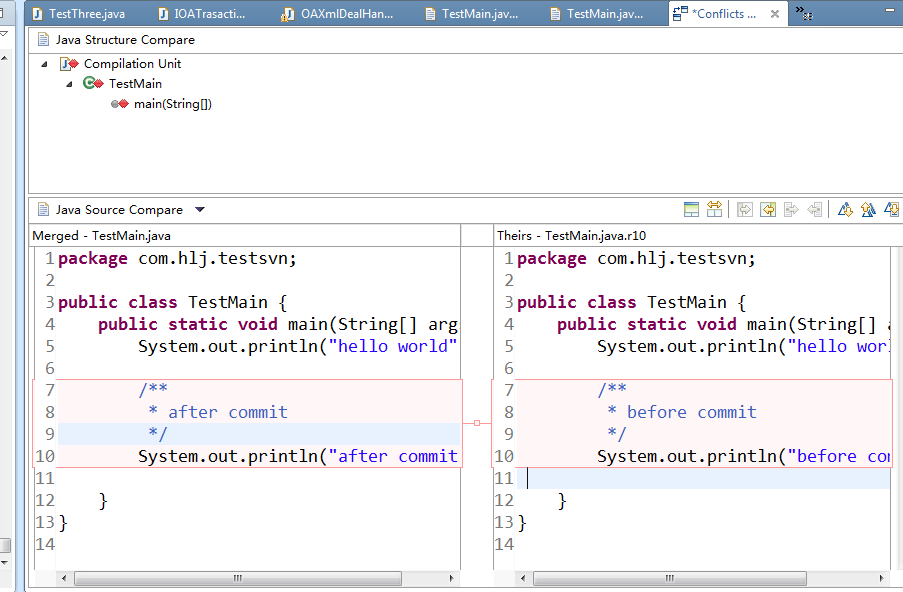
### 4、r10文件为，他先上传的文件，也就是对方本地目前的文件

|  |
| --- |
| public class TestMain {  public static void main(String[] args) {  System.out.println("hello world");  /\*\*  \* before commit  \*/  System.out.println("before commit ");  }  } |

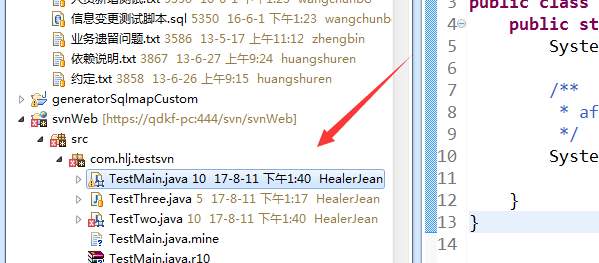
### 5、显示对比，点击冲突的文件右键->编辑冲突



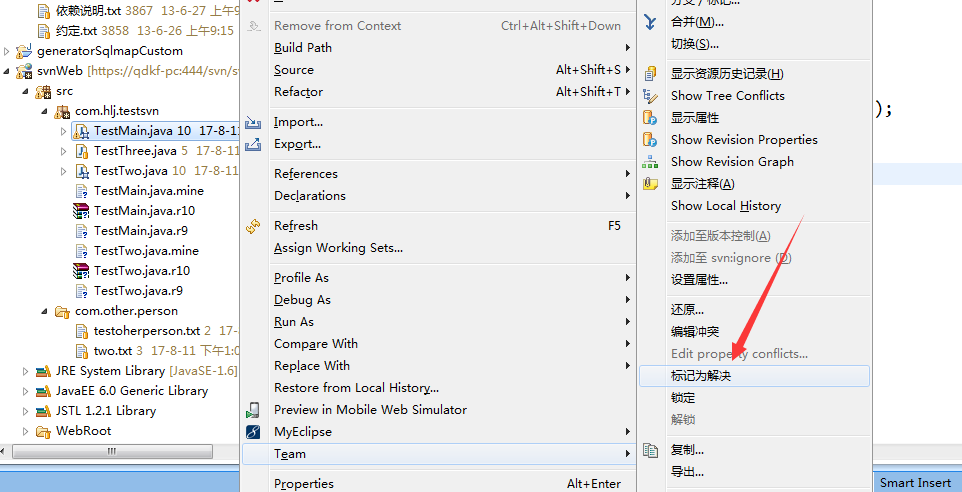
#### 5.2显示，可以编辑本文件，解决冲突，然后保存



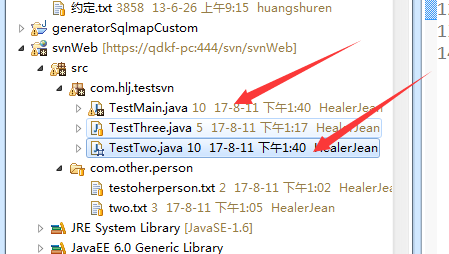
#### 5.3。保存，发现没有报错了



### 6、去除多出来的文件，选择原冲突文件（这里已经解决了）右键->标记为解决



#### 6.2 多出来的文件清除了，不在了



## 3、解决冲突方法总结

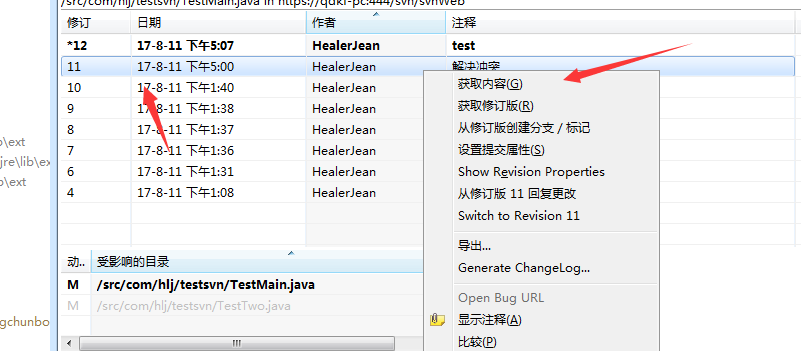
### 1、利用上面的5、6

### 2、后缀进行保存一个，不利用svn工具

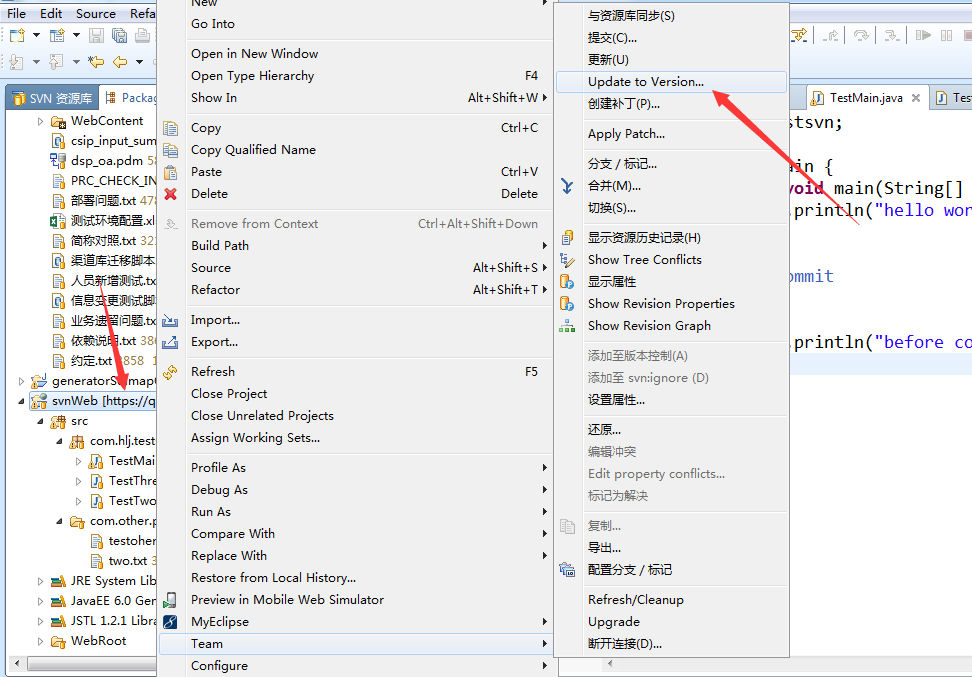
### 3、冲突文件直接修改，不利用svn工具

# 7、回退,尽量不要用全部，只退某个文件好

## 1、单个文件回退版本，本文件显示历史资源，然后选择版本，右键获取内容，即可回退本文件版本



## 2、版本全部回退，选中项目，右键->update to version



## 8、恢复刚刚删除的文件，右键->回复

