

目 录

- SVN相关的基本概念
- TSVN的使用流程及图标说明
- TSVN基本功能的介绍
- Eclipse下SVN的使用

TortoiseSVN的安装

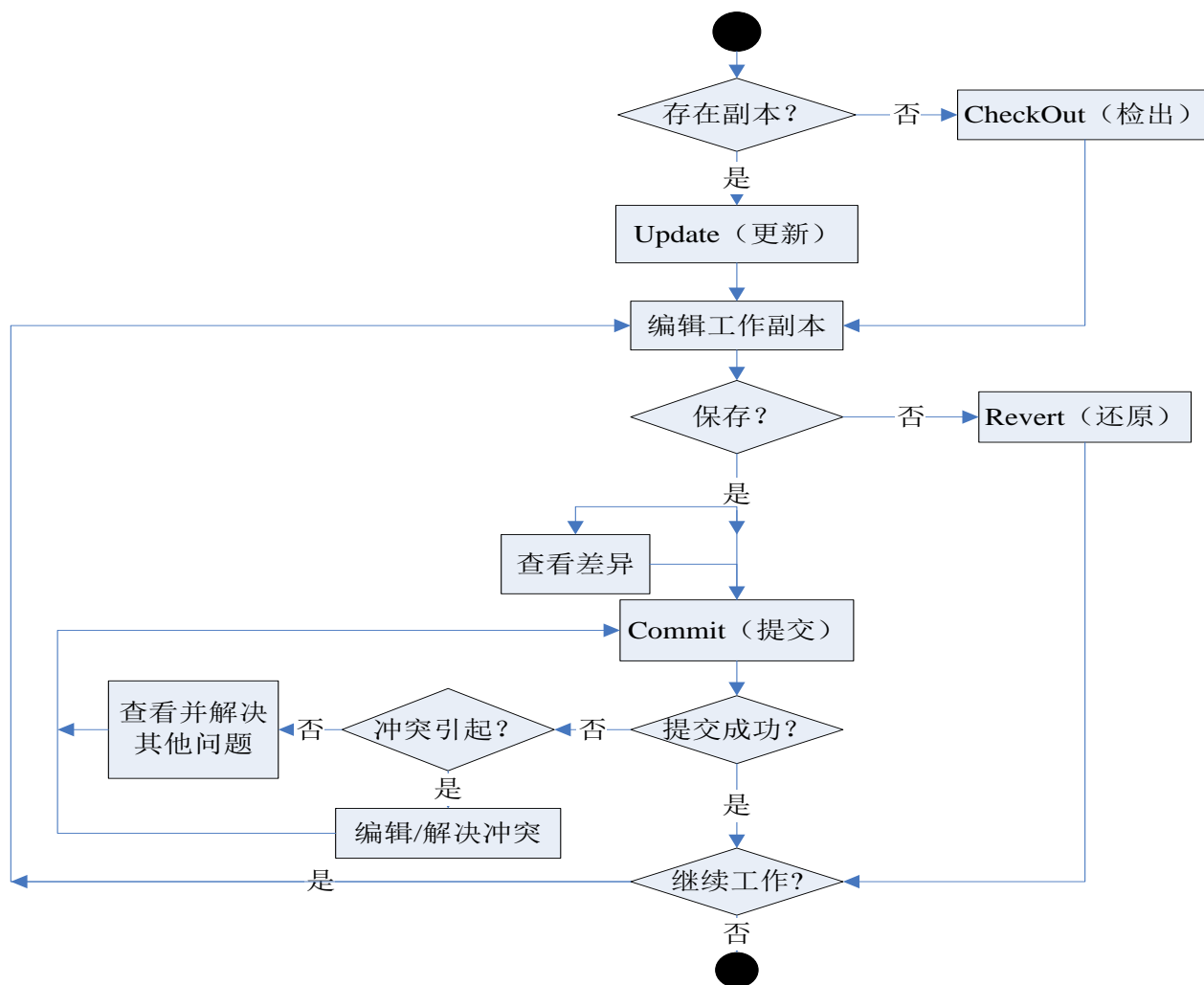
- 需要安装客户端



- 安装完毕后，在“资源管理器”里点击右键，会有如下菜单出现：



——TSVN的使用基本流程



——SVN使用图标说明



一个新检出的工作副本使用绿色的对勾做重载。
表示Subversion状态正常。



当我们开始编辑一个文件后，图标将变成红色感叹号。通过这种方式，可以很容易地看出我们对哪些文件进行了修改操作，但是还没有提交到版本库中；





如果在提交的过程中出现了冲突，图标将变成黄色感叹号。



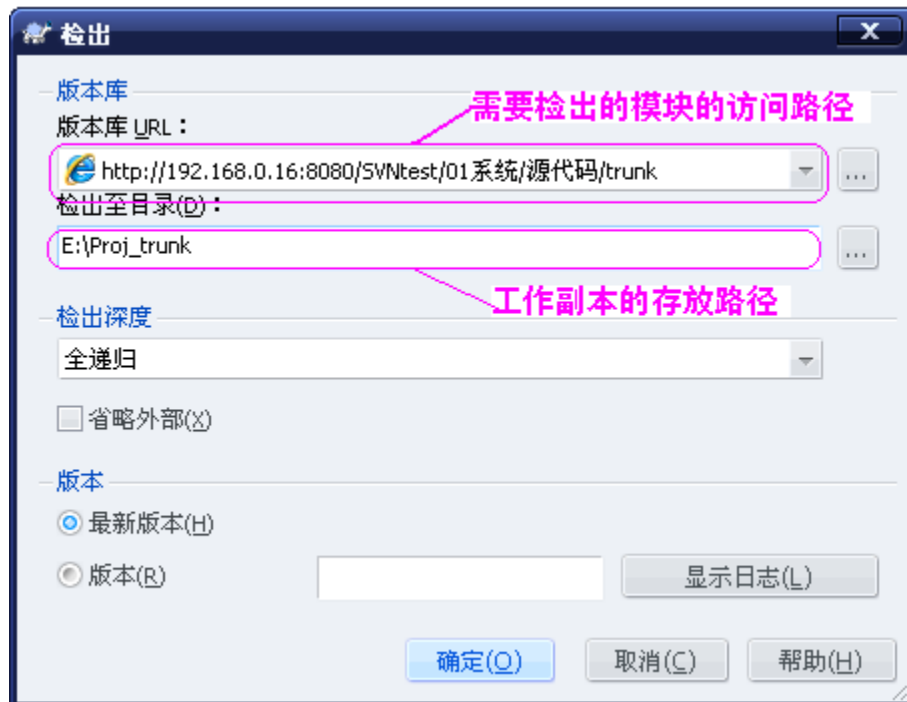
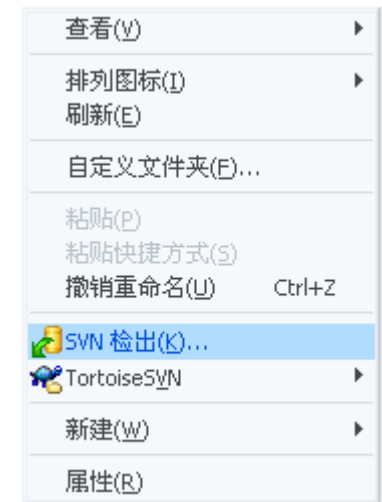
如果我们拥有了一个文件的锁，并且Subversion状态是正常，这个重载图标将提醒我们：如果不使用该文件的话，请进行释放锁操作，允许其他成员提交对该文件的修改。

——SVN使用图标说明

-  这个图标表示当前文件夹下的某些文件或文件夹已经被计划从版本控制中删除，或是该文件夹下某个受控的文件丢失了。
-  加号告诉我们有一个文件或目录已经被计划加入版本控制。

CheckOut(检出)

- 作用：将版本库中的内容检出到本地工作副本
- 步骤：
 - 1.新建一个空文件夹； 比如：E:\Proj_trunk
 - 2.在此目录中点击右键-> SVN Checkout...



CheckOut的注意事项

■ 检出深度：

1. 全递归(默认选择)

检出完整的目录树，包含所有的文件或子目录。

2. 直接节点，包含目录

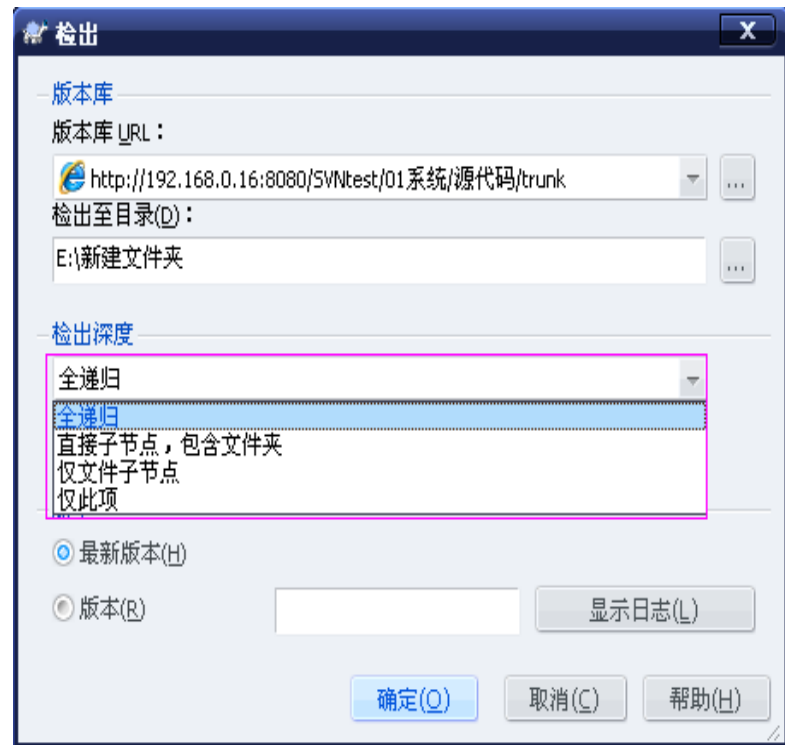
检出目录，包含其中的文件或子目录，但是不递归展开子目录。

3. 仅文件子节点

检出指定目录，包含所有文件，但是不检出任何子目录。

4. 仅此项。

只检出目录。不包含其中的文件或子目录。



省略外部引用：如果项目含有外部项目的引用，而这些引用我们
不希望同时检出，请选中忽略外部项目复选框。如果选中了这个
复选框，更新的时候要使用命令”更新至版本Update to
Revision...”

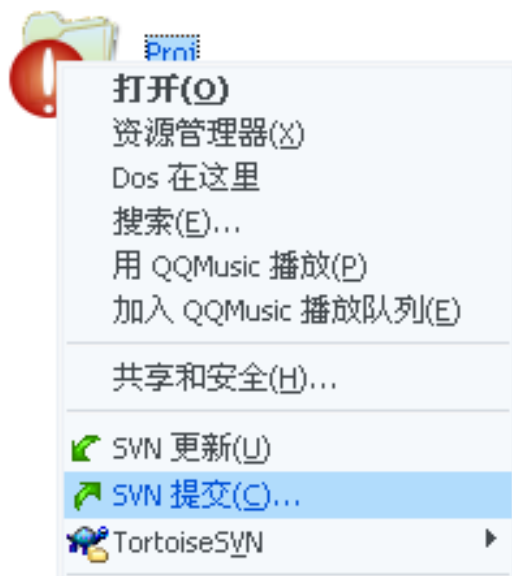
Update(更新)

- 作用：更新工作副本使其成为版本库中的最新版本
- SVN将显示出更新的文件和更新的次数



Commit(提交)

- 对工作副本进行编辑后提交到SVN
- 在右键菜单中点击SVN Commit
- 提交前写好信息，点击确定



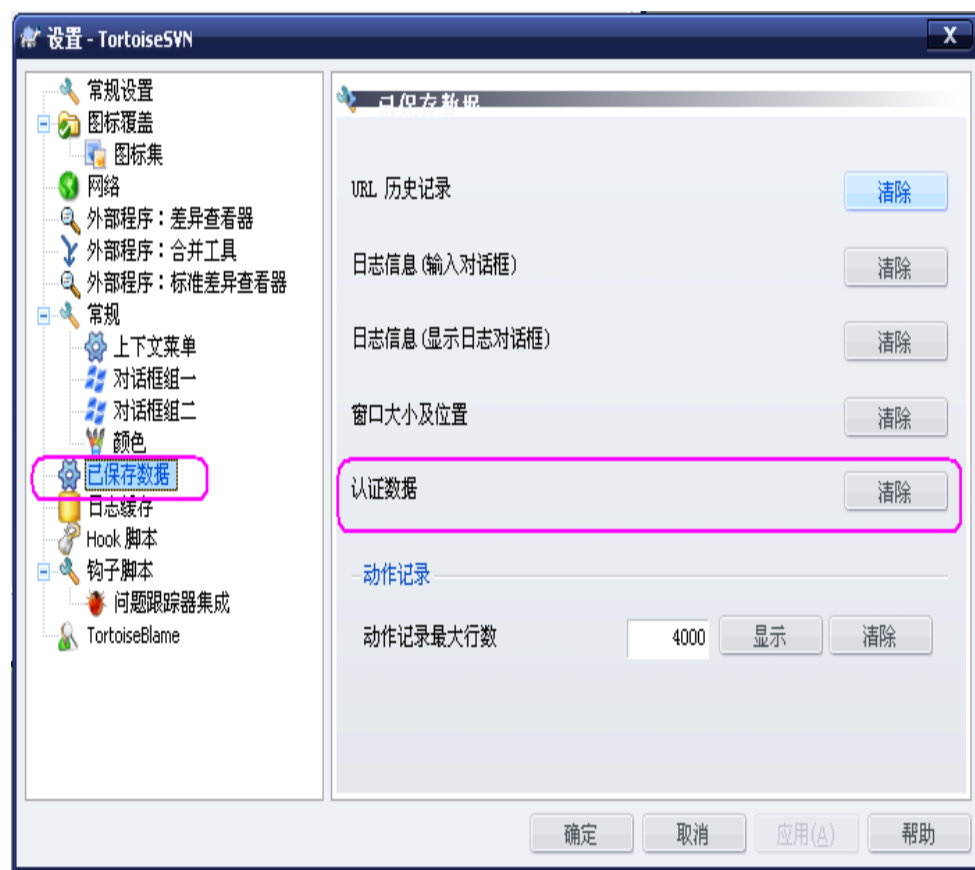
——权限控制

- 当进行提交文件操作的时候
您将看到权限提示信息
- 输入您的用户名和密码
- 保存权限设置（见红圈），
可以避免将来重复输入用户
名和密码

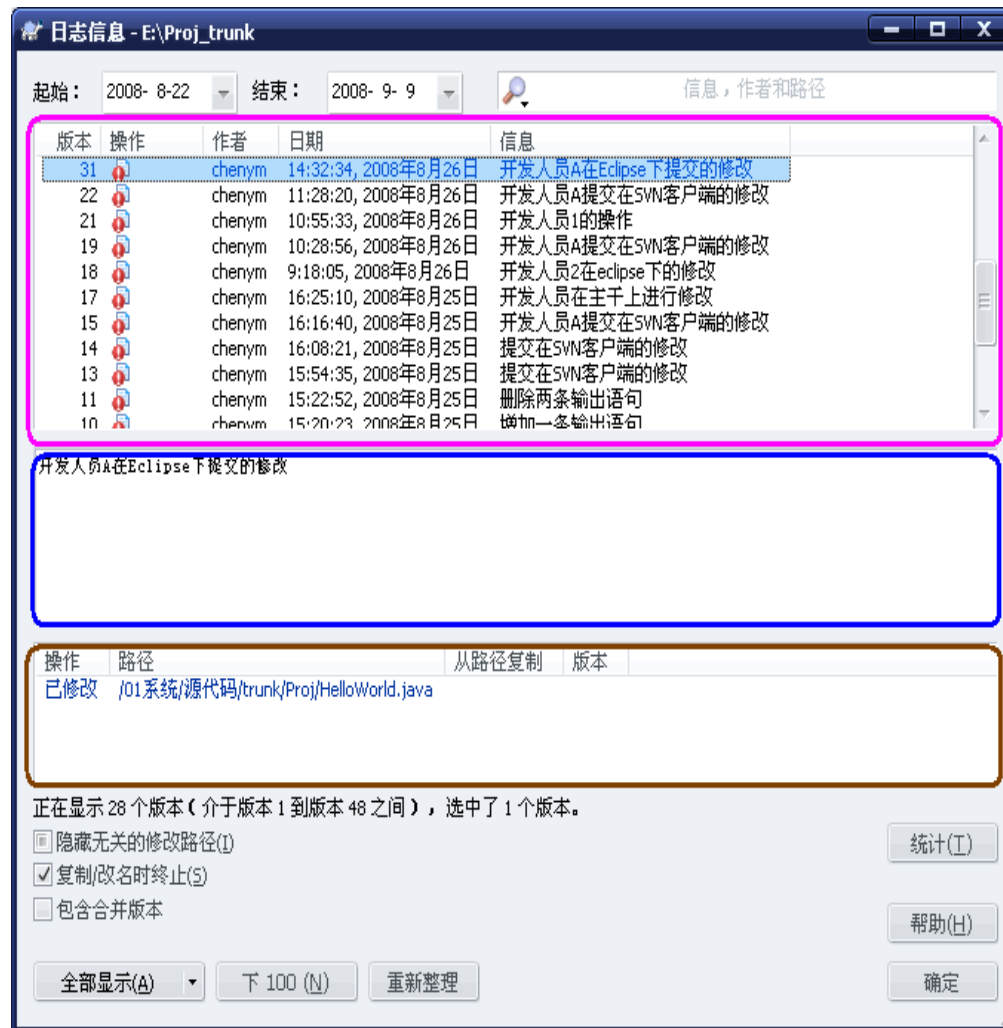


——如何删除认证数据

步骤： 点击右键 → 选择设置 → 以保存数据 → 清除认证数据



Show log (显示日志)



显示版本列表, 包含了日期和时间, 以及提交的用户和日志信息开头的部分内容。

显示了被选中的版本的完整的日志信息

显示了被选中版本中都对哪里文件和文件夹进行了修改。

——日志信息填写规则

■ 好的日志信息和糟糕的日志信息

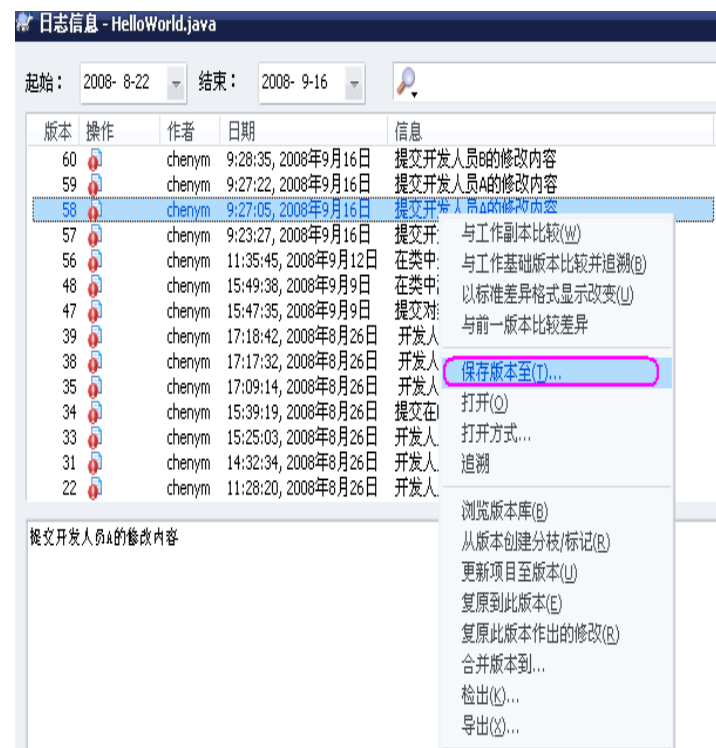
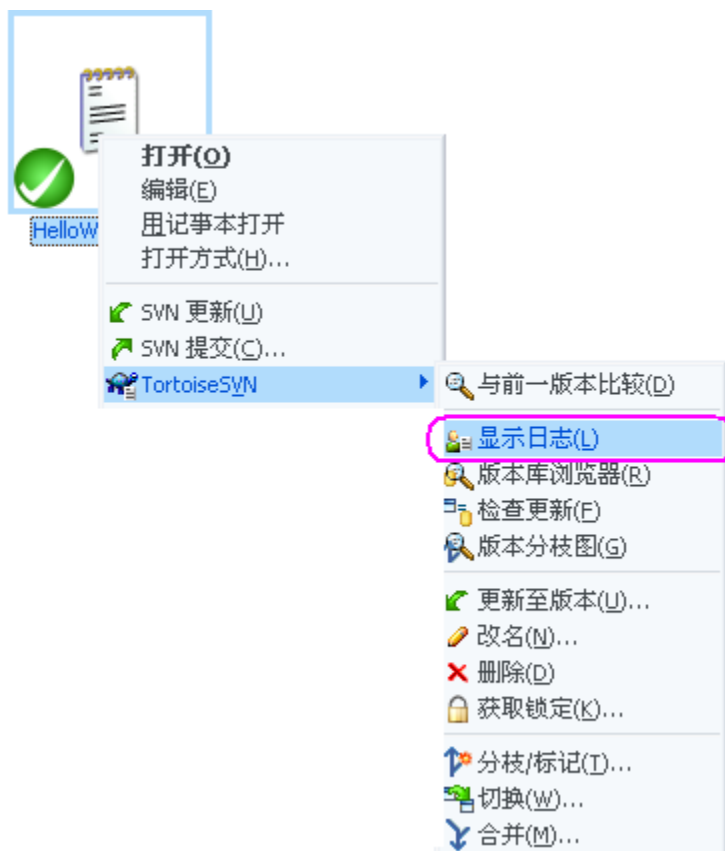
日志信息主要记录的是每次的修改内容。建议把一些重要数据、关键操作写到日志信息中。

- **注：**修改人和提交时间由软件自动记录，无需人工写入日志信息

不规范的日志	规范的日志
去除无用文件	删除分支中的无用文件attachment.js_BAK_和moderation.js”
线索报警	修改线索的样式
更新配置文件	更新公司主题包和栏目配置文件

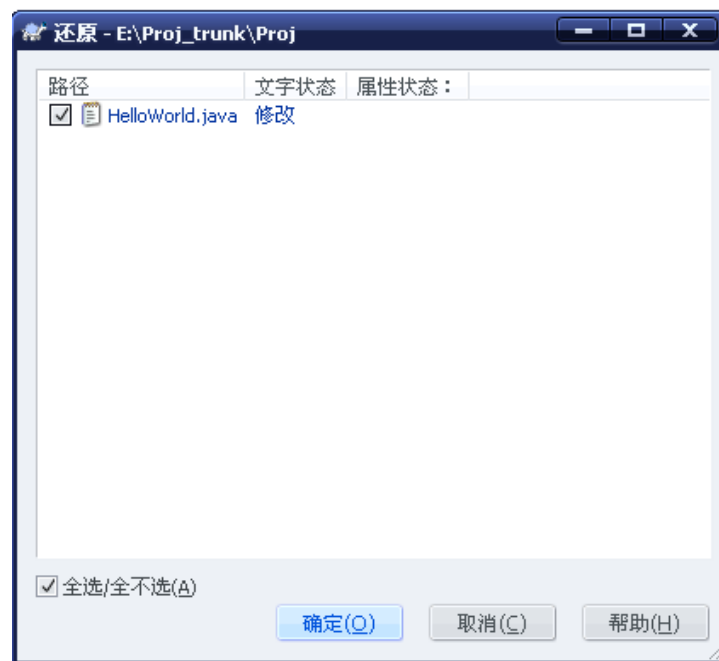
——如何得到历史版本

- 工作副本右键→显示日志→选择所需的版本号→保存版本至



Revert(还原)

- 作用：撤销本地所有未提交的修改
- 注意：还没有执行Commit操作之前执行此命令才可以，否则无效



ADD(添加)

- 选中文件/文件夹（在新文件/文件夹所在父文件夹点击右键），
- 在菜单中选择“添加Add”命令。不需要受SVN控制的文件请取消打钩。



Add to Ignore List （忽略文件）

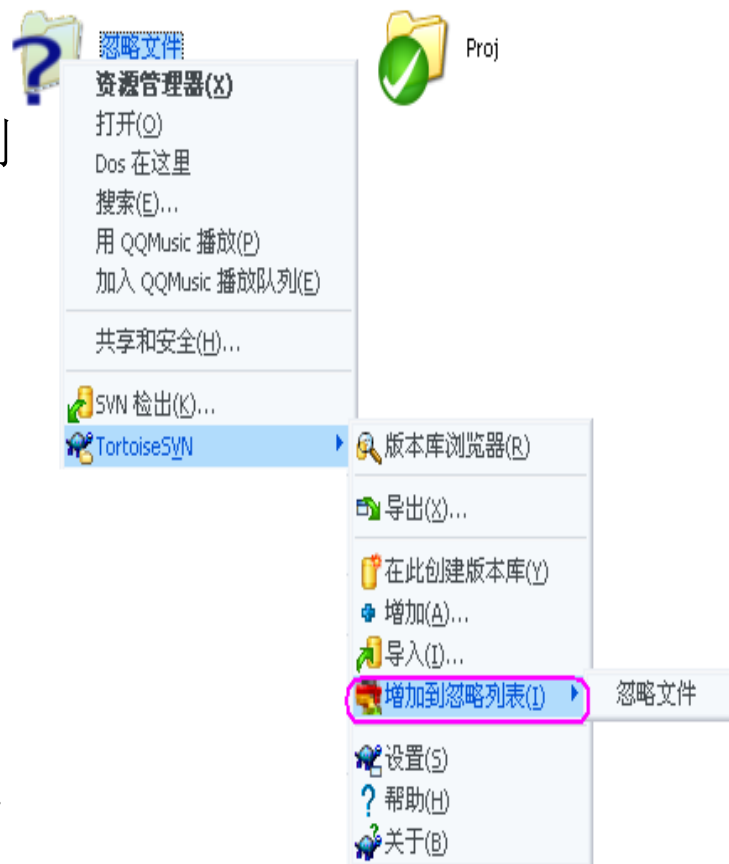
■ 添加忽略文件

右键一个单独的未进入版本控制文件 → TortoiseSVN → (加入忽略列表) Add to Ignore List, 会出现一个子菜单允许你仅选择该文件, 或者所有具有相同后缀的文件。

■ 删除忽略文件:

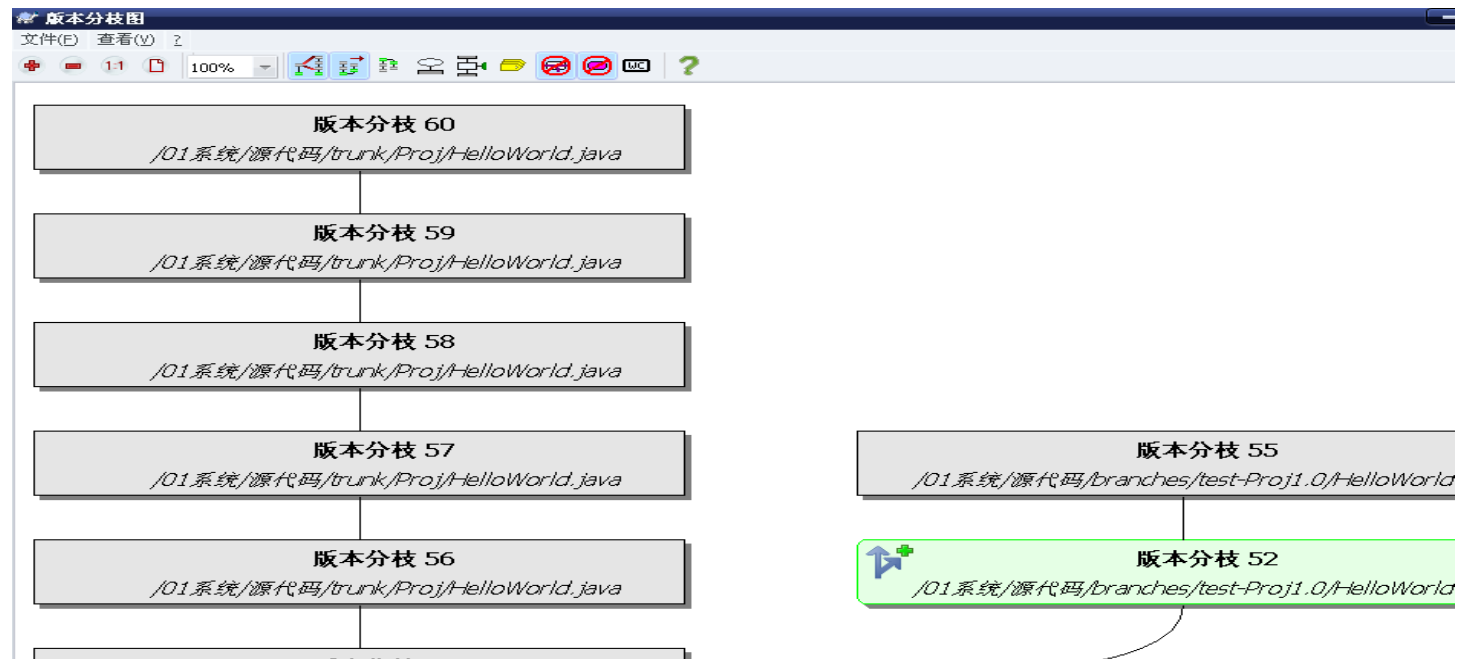
如果你想从忽略列表中移除一个或多个条目, 右击这些条目, 选择TortoiseSVN → 从忽略列表删除。

■ 已进入版本控制的文件或目录不能够忽略



版本分支图

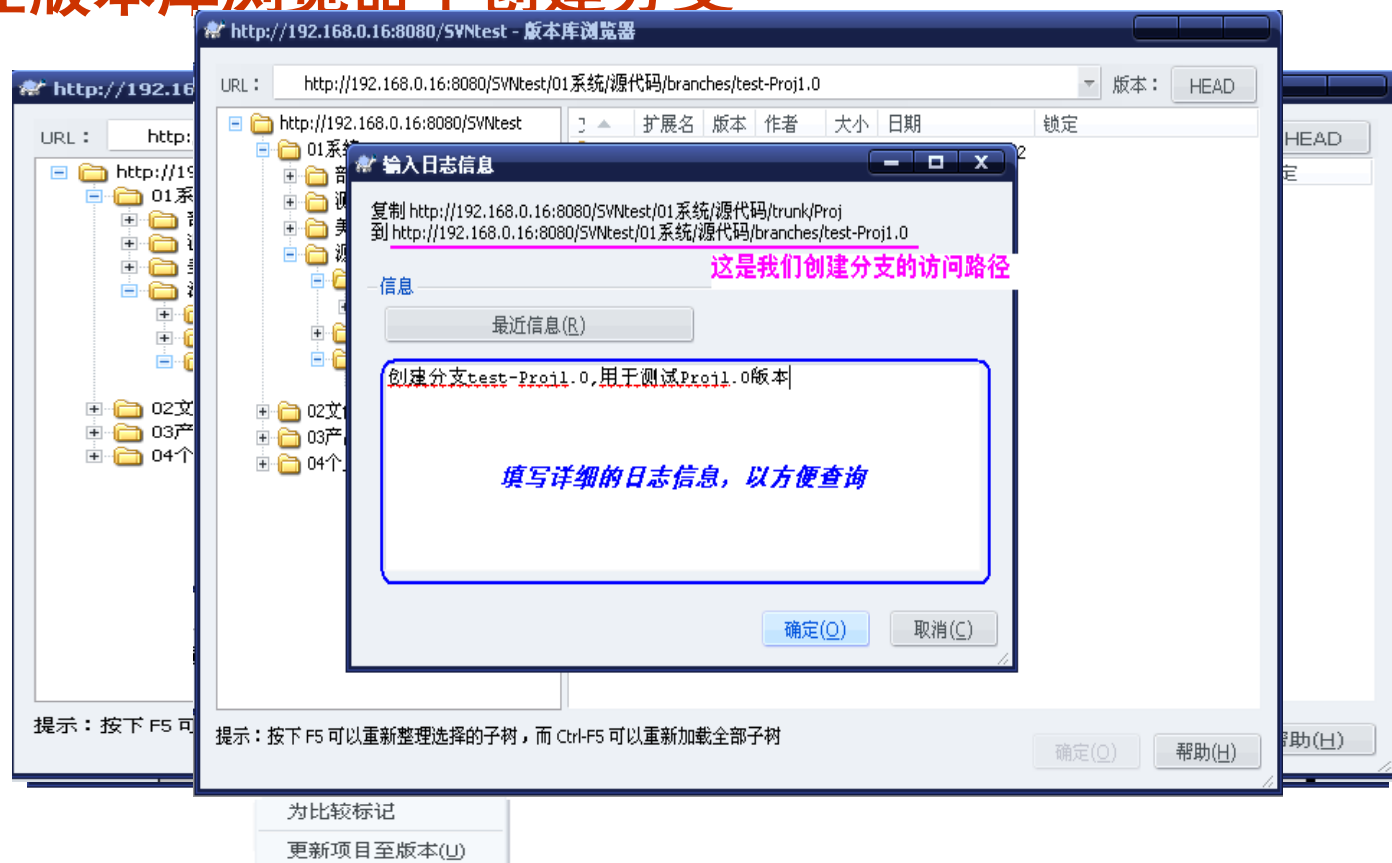
- 这个版本历史分析图能够显示分支/标签从什么地方开始创建，以及什么时候删除。
- 版本分支图将显示以下内容：
 - (1) 增加文件/文件夹 (2) 已删除文件/文件夹
 - (3) 分支最新版本 (4) 一般的文件/文件夹



如何创建分支 (Branch)

■ 两种方式:

1. 在版本库浏览器中创建分支

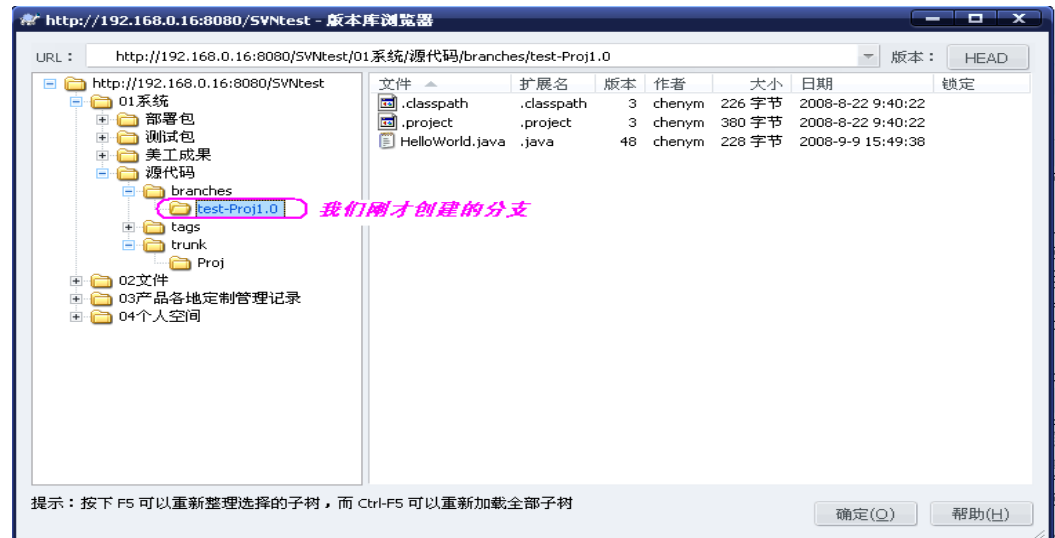
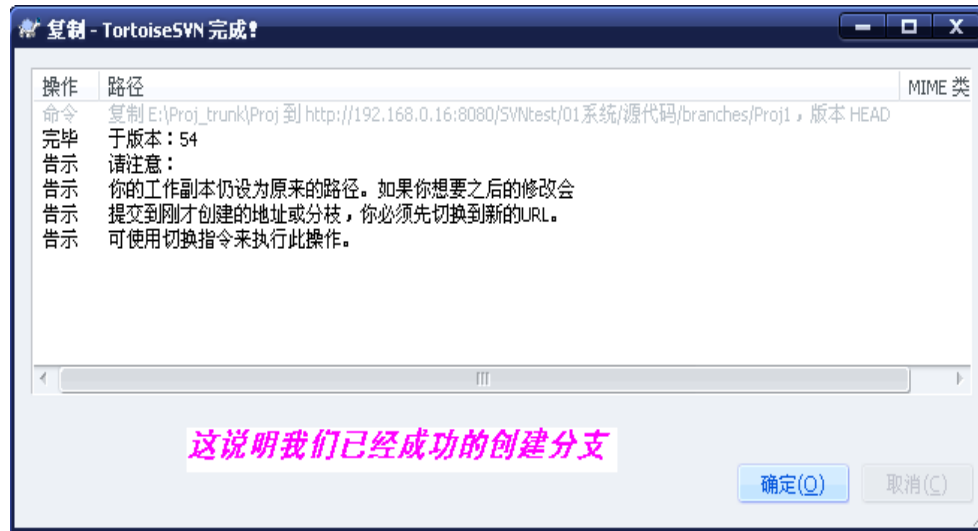


如何创建分支 (Branch)

2.在工作副本创建分支



如何创建分支 (Branch)



Switch(切换)

- 文件创建分支后，可通过Switch选择在主干工作或者在分支工作；

- **注意：**

(1) 切换操作起来就象更新，因为它没有丢弃我们在本地做的修改。

(2) 在工作副本里，当我们进行切换的时候，任何没有提交过的修改都会被混合。

- **解决方法：**

- 1) 在切换前提交修改；
- 2) 把工作副本恢复到一个已经提交过的版本(比如最新版本)。



合并

- 假设我们在分支上进行修改：添加一条输出语句：
System.out.println(“测试是否能将分支的修改合并到主干中!”) ;提交到配置库中，现在执行将分支合并到主干的操作

```
public class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello World!") ;
    }
}
```

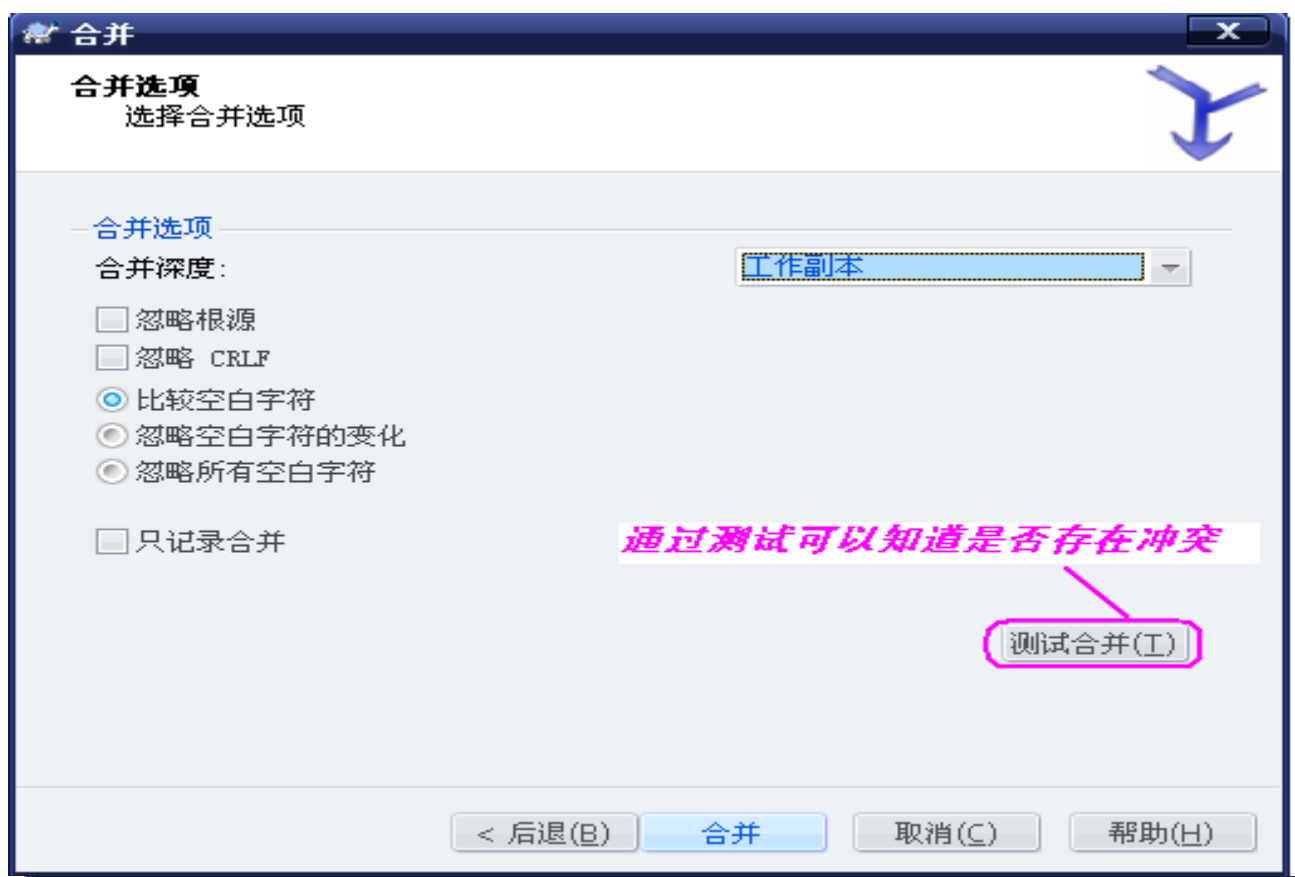
这是HelloWorld的主干内容

```
public class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello World!") ;
        System.out.println("测试是否能将分支的修改合并到主干中!") ;
    }
}
```

这是HelloWorld分支的内容

分支合并到主干

- 右键主干的文件→选择合并

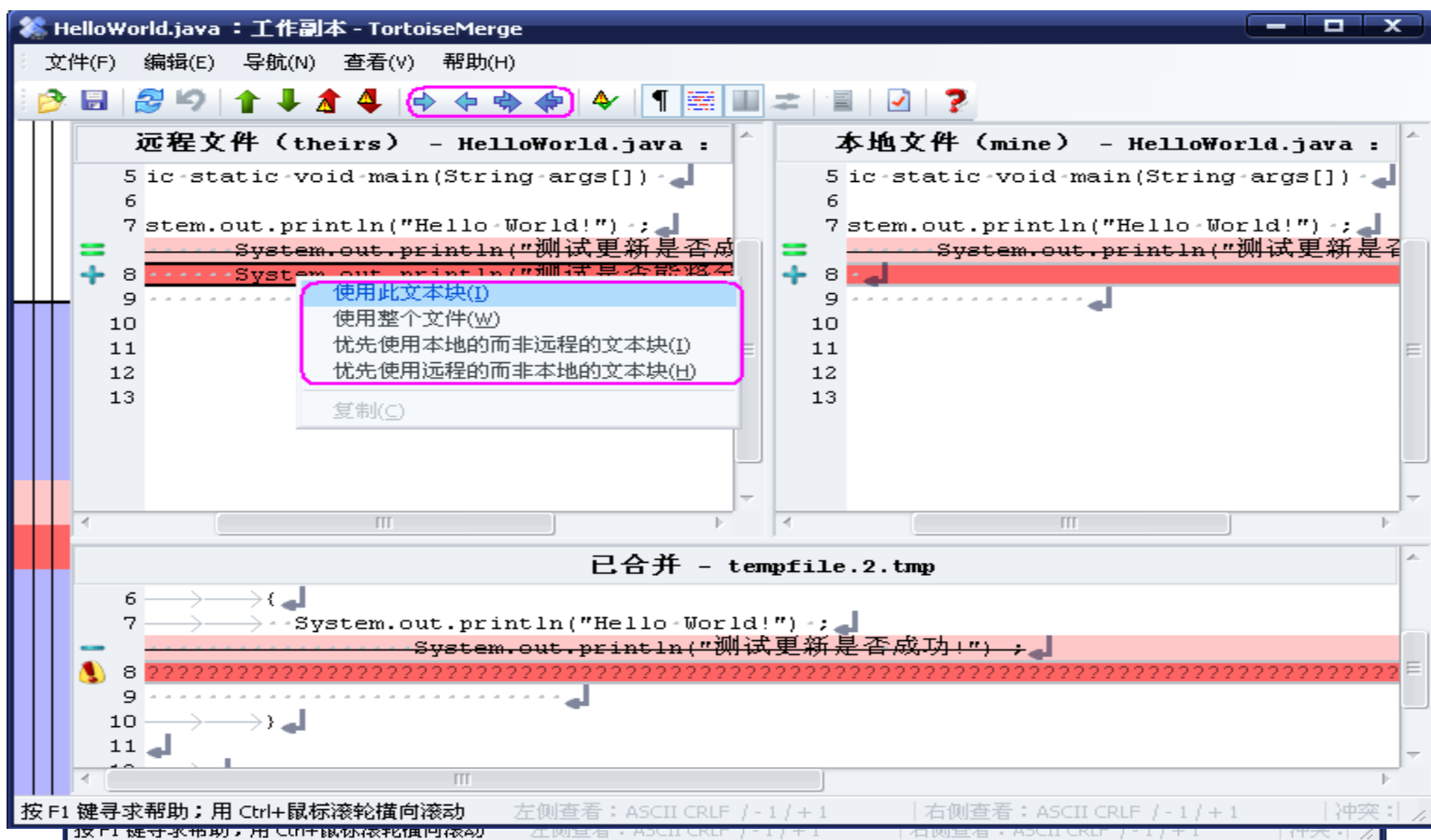


导出(X)...
重新定位(O)...

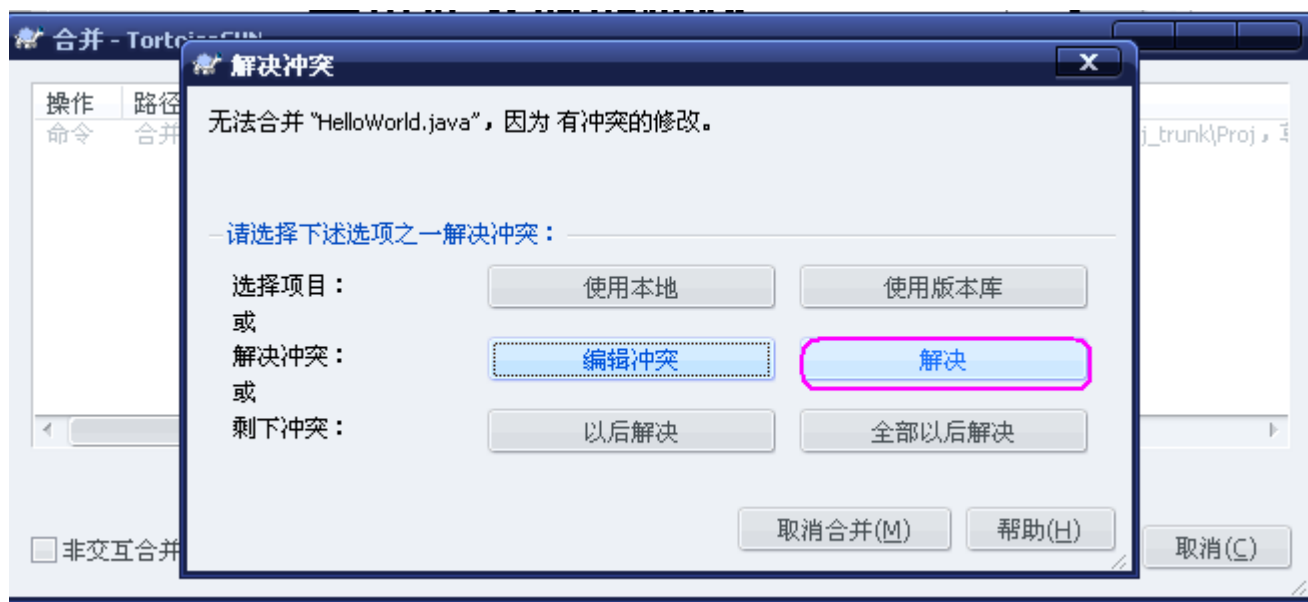
合并时出现冲突



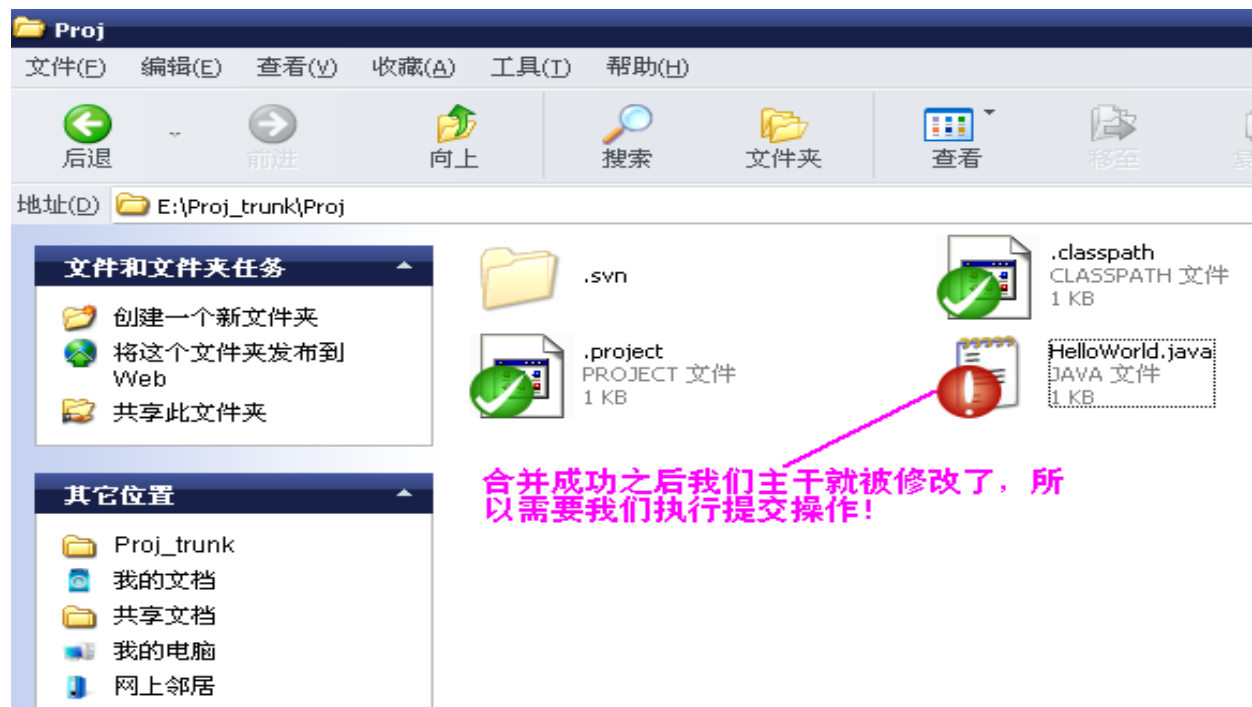
编辑冲突



冲突已解决



合并成功



冲 突

- 什么是冲突
- 冲突产生的原因
- 冲突产生的时机
- 解决冲突
- 关于避免冲突的建议

什么是冲突

■ 冲突是指

团队协作工作时，当多位团队成员同时修改同一个文件，造成本地文件与SVN系统中的文件版本不一致，而导致文件无法提交的情况

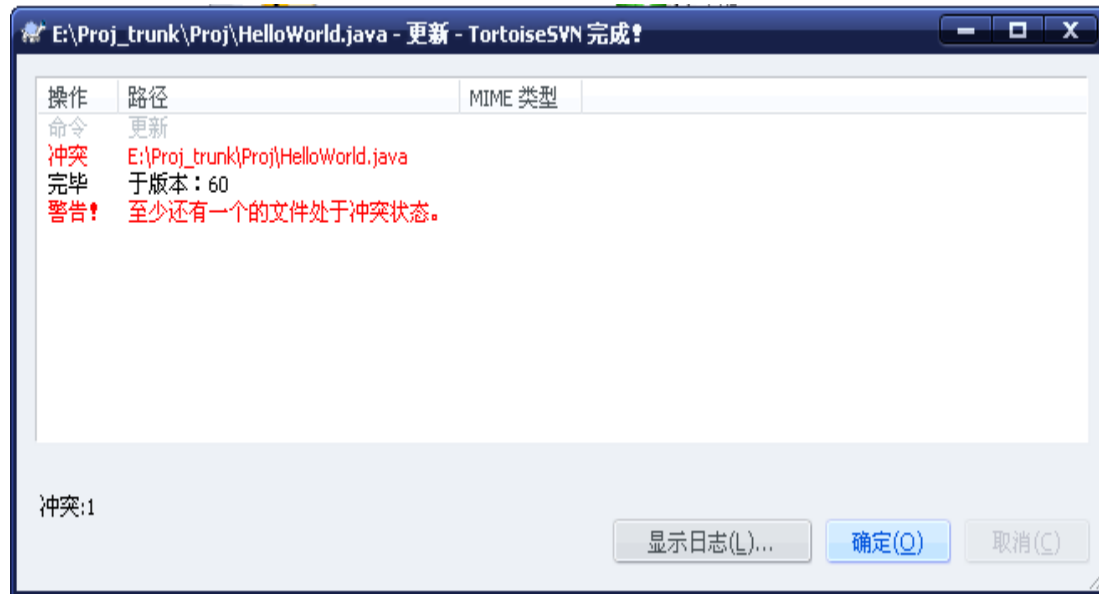
冲突产生的原因

- 产生原因

当团队协同工作的时候，多位团队成员同时操作一个文件。团队成员A操作完成后，将该文件提交到SVN上。此时，其他团队成员的本地文件与SVN上的文件版本不一致。当团队成员B操作完成并对文件进行提交操作时，就会产生冲突。

冲突的提示页面

■ 提交时产生的冲突



冲突的解决方案

遇到冲突时，需要update该文件。我们将看到三个带问号的文件

filename.mine

filename.rOLDREV

filename.rNEWREV

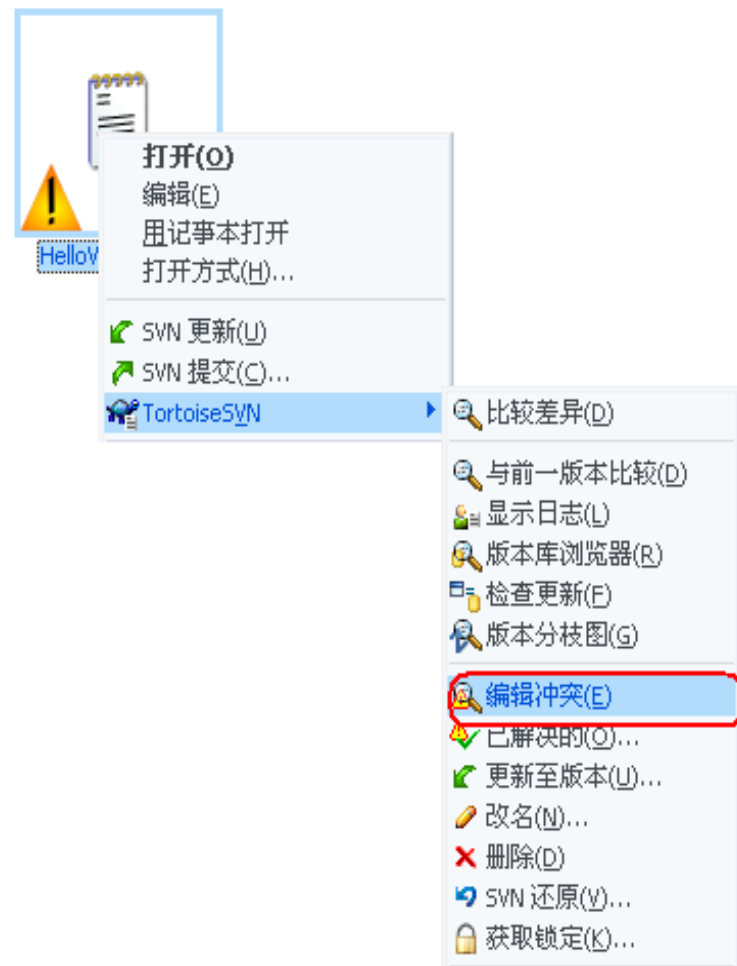


面对文件冲突，我们可以选择以下两种方式解决冲突

- 使用工具解决冲突
- 用revert放弃所做的修改

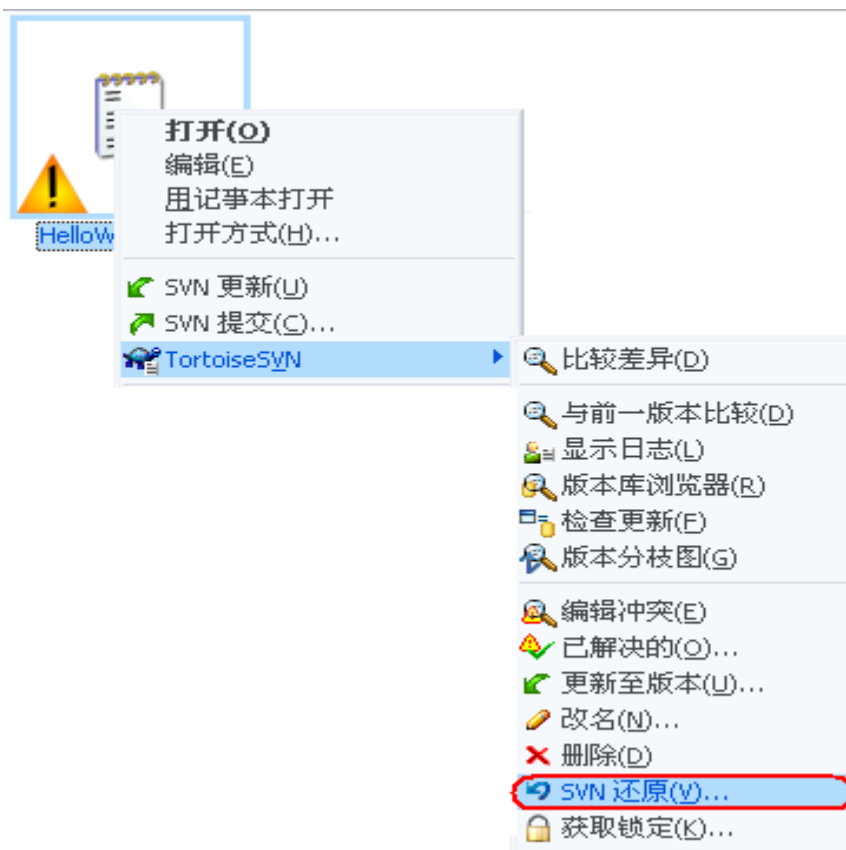
使用工具解决冲突

- 使用自带工具，如图所示
- 对比两个版本的文件，解决冲突
- 工具中会列出两个版本冲突的部分，并让我们选择使用哪个版本的内容



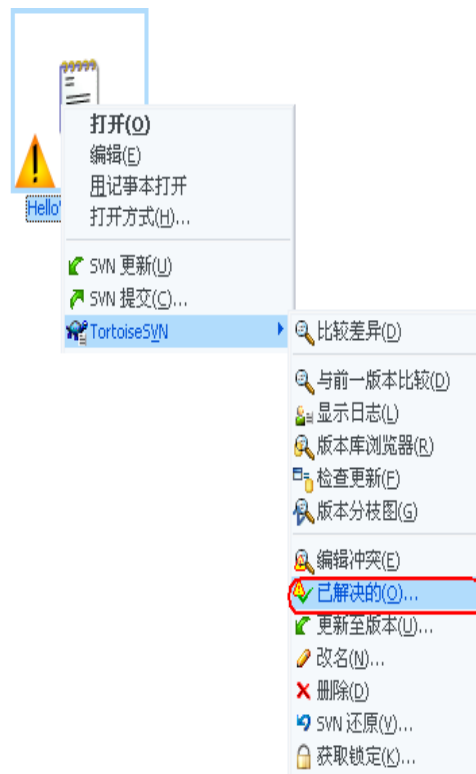
用revert放弃所做的修改

- 选择SVN还原（revert） 放弃自己所做的修改



解决冲突后

- 通过上面所说的方法，解决冲突后要选择已解决的（resolved）
- 解决后，带问号的三个文件将自动删除
- 最后选择commit提交到SVN



关于冲突的建议

- 修改文件之前，先进行一次update操作
- 修改完成后，及时commit，不要在本地停留过长时间
- 在多位团队成员协作时，尽量修改自己撰写的部分，尽量不要修改不属于自己撰写的部分
- 出现冲突很正常，可以通过前面的方法解决，不要相互覆盖

其他SVN的功能

- **Export（导出）**

导出你需要的文件，导出后不在受SVN控制；而checkout检出的文件仍受SVN控制；

- **Import（导入）**

将需要的文件导入到版本库中；

- **Relocated（重新定位）**

当服务器上的版本库目录已经改变，我们可以把工作副本重新定位；




- **CleanUp（清理）**

递归的清理工作副本，删除锁定和保留的未完成操作，如果你得到工作副本已经锁定的错误，运行这个命令删除无用的锁定，并将工作副本恢复到可用的状态；

Subclipse的安装步骤

1. 帮助 > 软件更新 > 查找并安装
2. 选择“要安装的新功能部件” 选项并点击Next。
3. 点击“新站点” 并且创建一远程站点，
名字:Subclipse
URL:http://subclipse.tigris.org/update_1.2.x
4. 在结果安装窗口中，把"Subclipse in the Features" 选择到安装列表中，并且通过向导来开始安装插件。
5. 完成这些之后，重新启动Eclipse。

在Eclipse下使用的图标

- 受SVN控制的文件被标记为小黄桶； 
- 修改过的文件都被标记上一个黑色星号； 
- 未受版本控制的文件被标记为问号； 

Eclipse下使用Svn的主要功能

- 在Eclipse下使用Svn我们主要使用功能:

1)将版本库导入到SVN资源库

2)将新建项目导入到版本库

3)小组

4)比较对象

5)替换为

——将版本库导入到SVN资源库

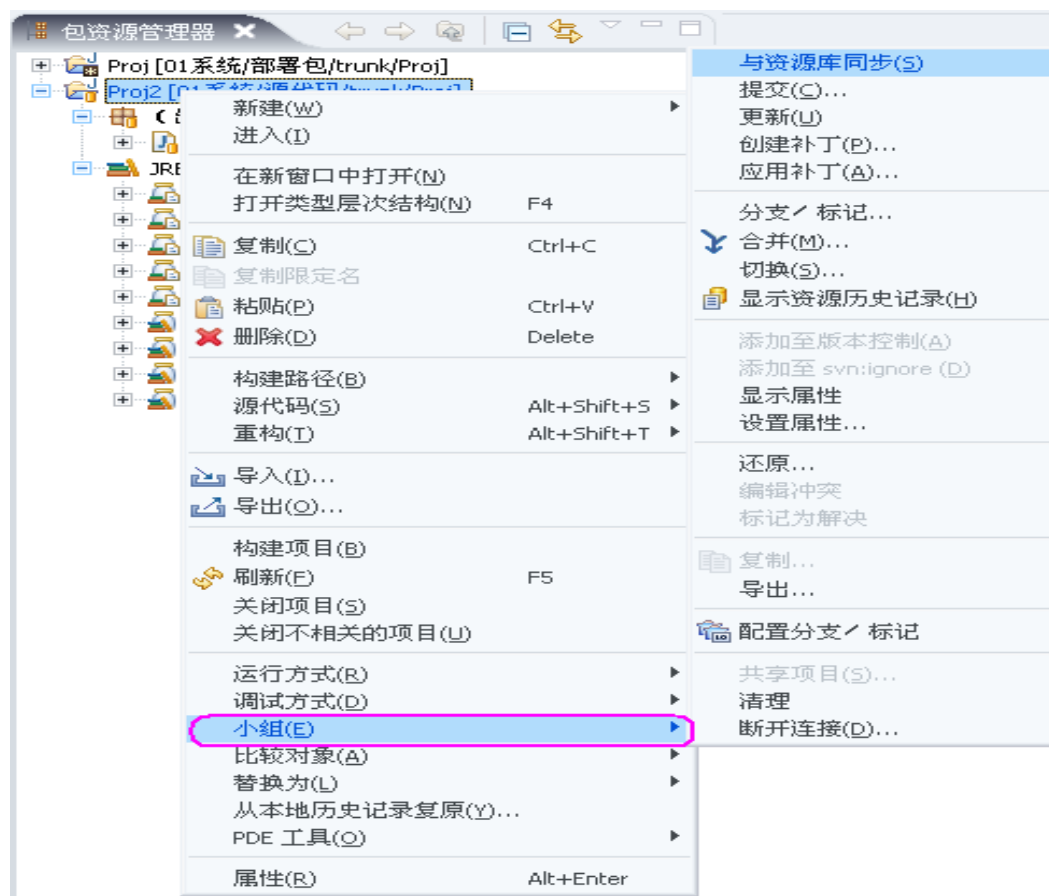


——将新建项目导入到版本库



Eclipse中小组的使用

■ 小组的主要功能(如图1.1)



比较对象



■ 例如：

This screenshot shows the Eclipse IDE with a file named 'HelloWorld.java' open. The code is as follows:

```
public class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello World!");
    }
}
```

Below the code, there is a pink text label: '修改前HelloWorld的内容' (Content of HelloWorld before modification).

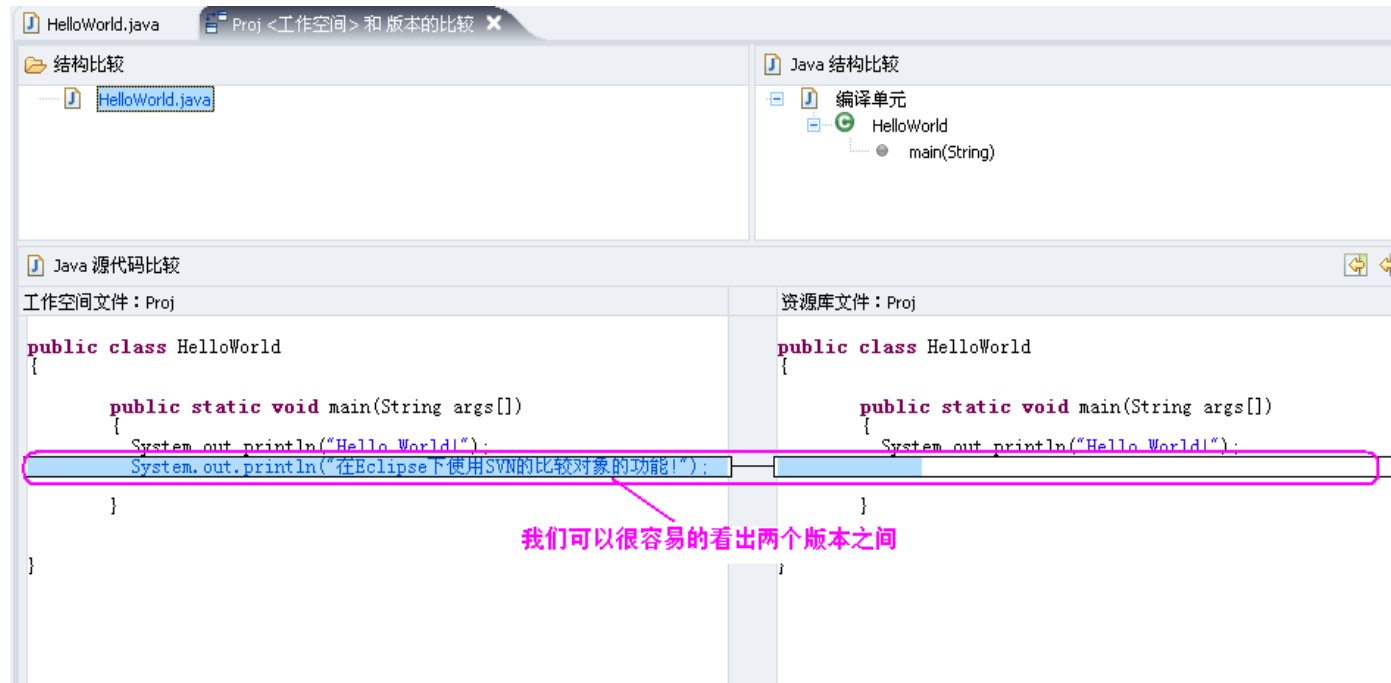
This screenshot shows the Eclipse IDE with the same file 'HelloWorld.java' open. The code is now modified:

```
public class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello World!");
        System.out.println("在Eclipse下使用SVN的比较对象的功能!");
    }
}
```

Below the code, there is a pink text label: '修改后HelloWorld的内容' (Content of HelloWorld after modification).

基本修订版

■ 1) 基本修订版:在Eclipse中的基本版

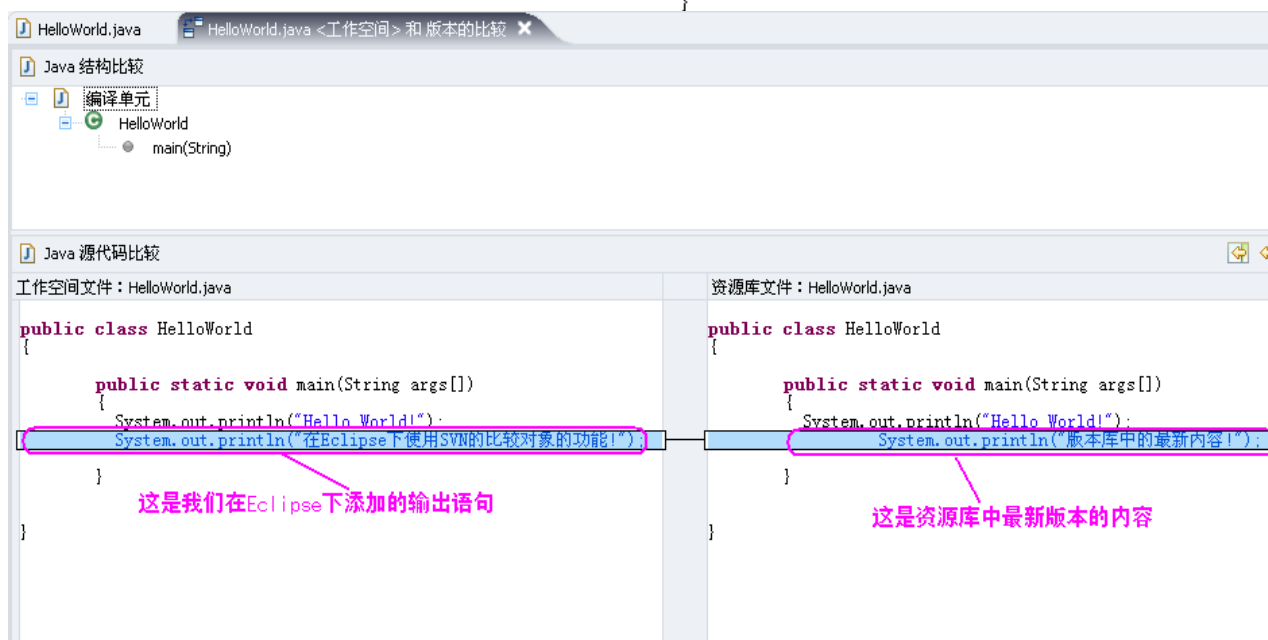


资源库中的最新内容

2) 资源库中的最新内容

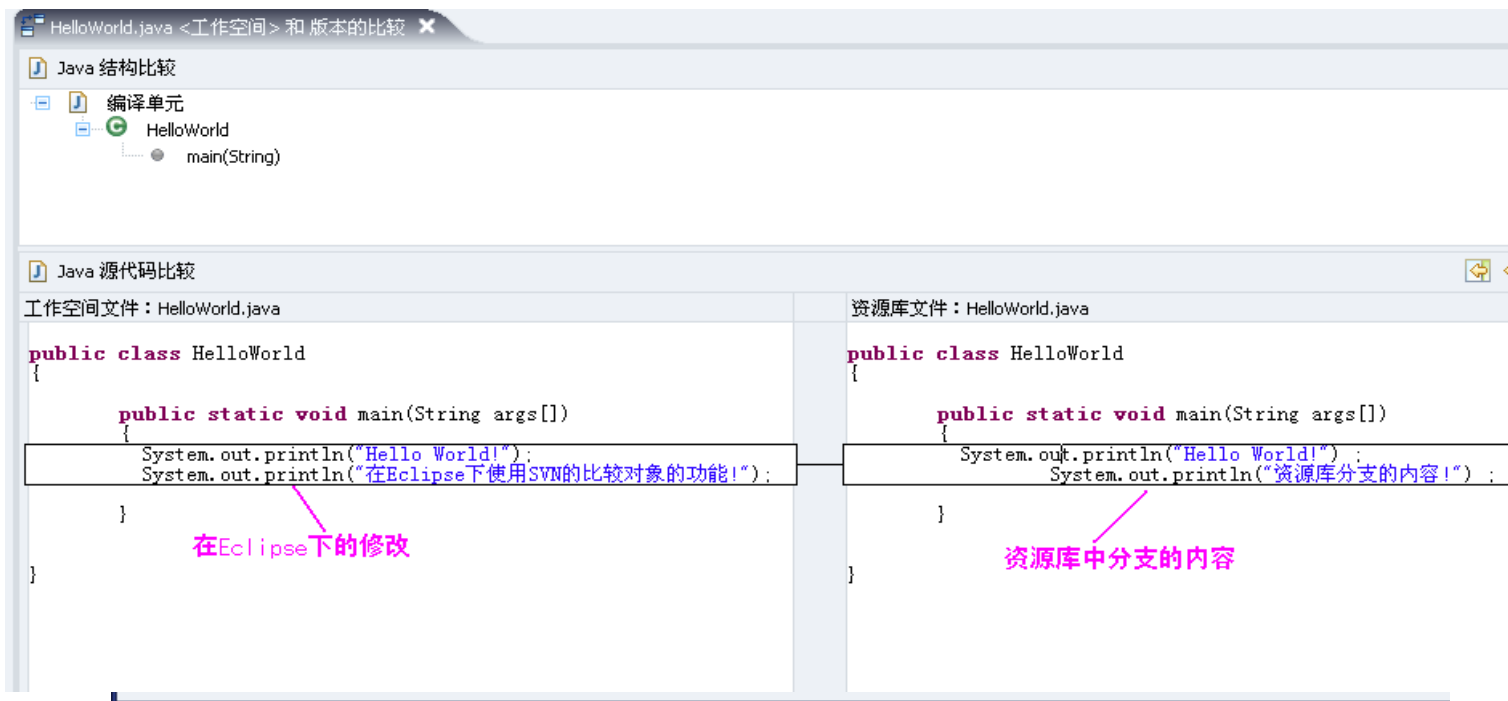
```
public class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello World!");
        System.out.println("版本库中的最新内容!");
    }
}
```

这是资源库中的最新版本



分支/标记

3) 分支/标记



修订版

4) 修订版

比较的对象是你选择的版本号对应的项目

The screenshot displays the Eclipse IDE's comparison interface for the file HelloWorld.java. It compares the current workspace version with revision 62.

Structure Compare Table:

修订	标记	日期	作者	注释
63		08-9-16 下午2:47	chenym	提交开发人员B的修改
*62		08-9-16 下午2:27	chenym	提交在Eclipse下的修改
60		08-9-16 上午9:28	chenym	提交开发人员B的修改内容
59		08-9-16 上午9:27	chenym	提交开发人员A的修改内容
58		08-9-16 上午9:27	chenym	提交开发人员A的修改内容

Java 源代码比较 (Java Source Code Compare):

Workspace file: HelloWorld.java

```
public class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello World!");
        System.out.println("在Eclipse下使用SVN的比较对象的功能!");
    }
}
```

Resource file: HelloWorld.java

```
public class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello World!");
    }
}
```

Annotations:

- 这是Eclipse工作台中的当前内容 (This is the current content in the Eclipse workspace)
- 这是版本号为62的内容 (This is the content of version 62)

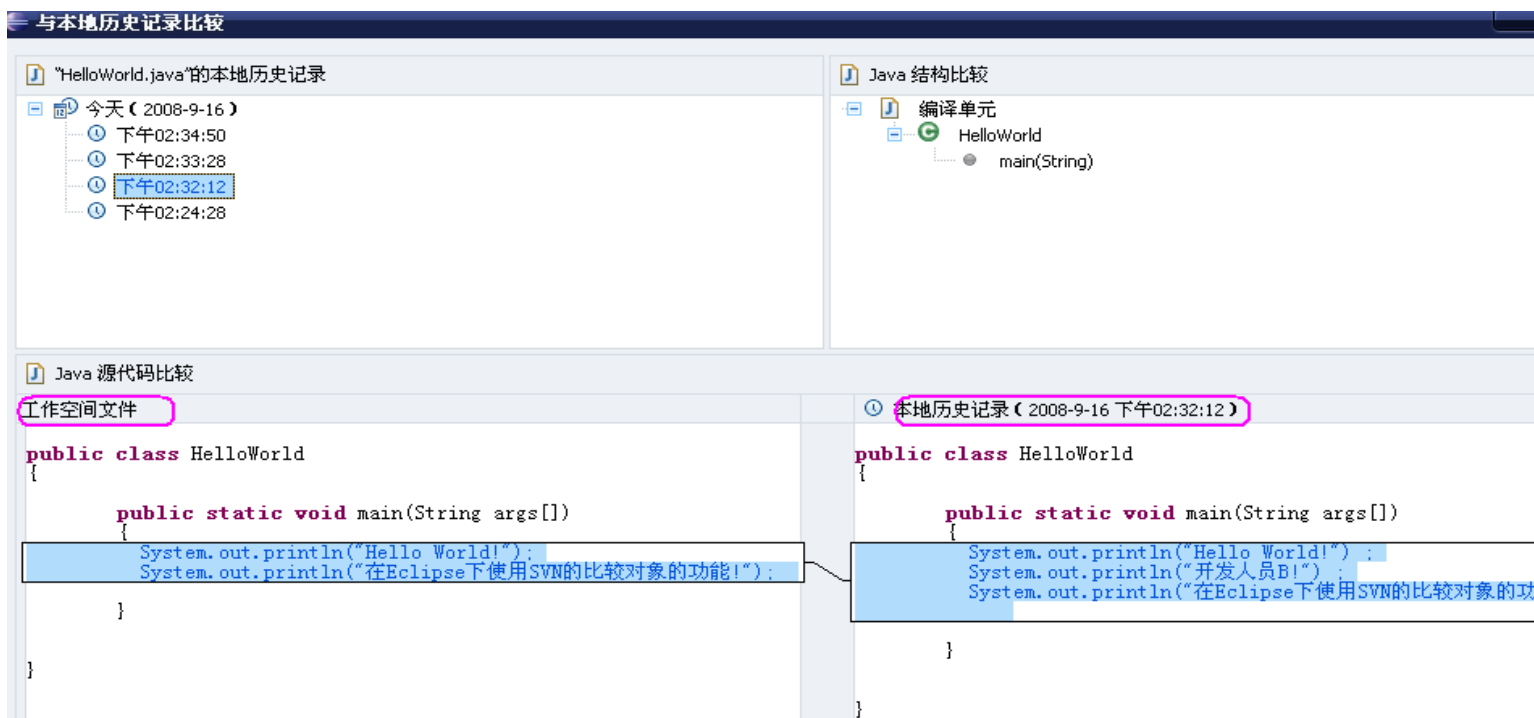
Java 结构比较 (Java Structure Compare):

- 编译单元 (Compilation Unit): HelloWorld
- main(String)

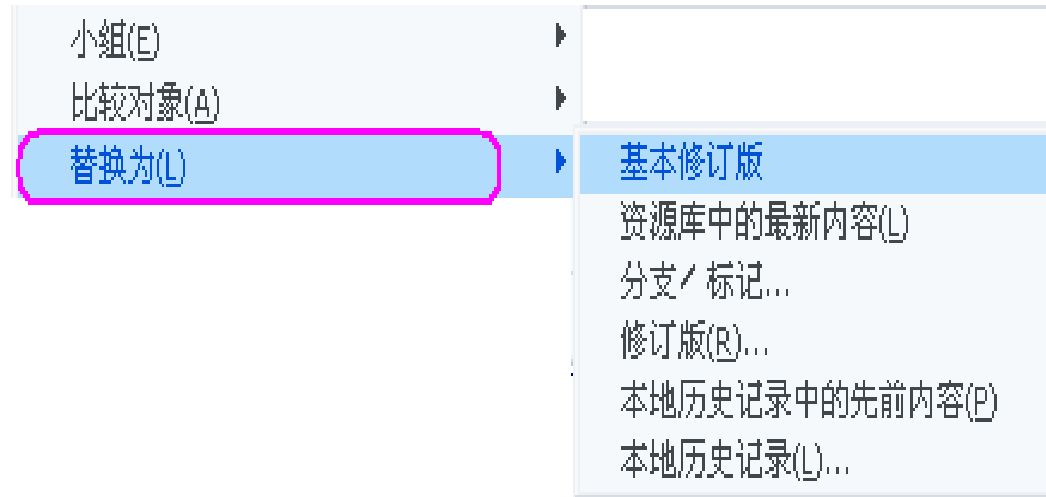
本地历史记录

5) 本地历史纪录

现修改的版本与本地修改的其他版本库进行比较



Eclipse中的替换为



注:与比较对象的应用一样,只是这是替换.

reference

- http://wenku.baidu.com/link?url=4QQ5HyU0aCCTN6iP9eIFRvoDzuiuEKvn8ne1FSZxh-nm-77AebUfFx9t8TJMFEIs0C3MprpFtX1cAAKhp-raMK1Q7Y5YyMjCXkZ9SjC6n_J__