### [Git的Merge合并冲突具体解决方法](http://www.cnblogs.com/wavky/p/3504060.html)

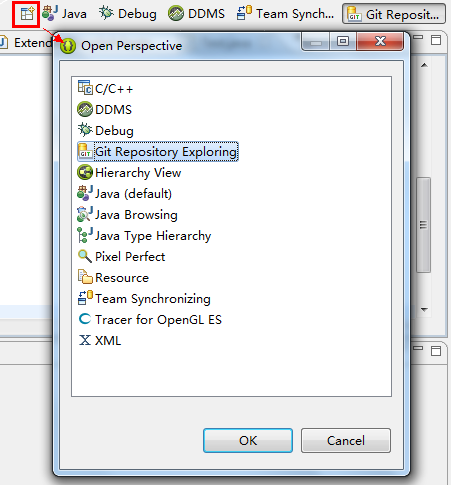
总结一下git的merge合并冲突解决方法。

**前提工作**

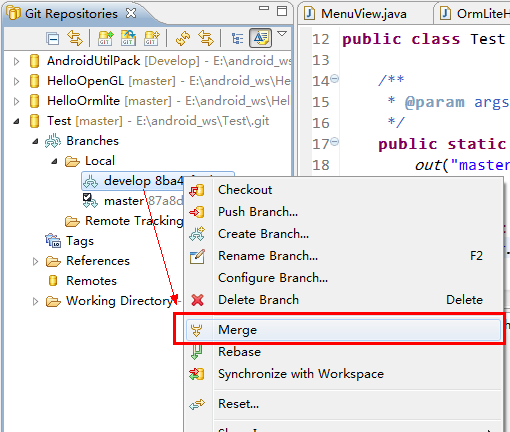
* 创建一个普通JAVA工程Test，创建一个类Test，写点东西并加入到本地git库；
* 创建个develop分支，创建个Develop类，修改Test类的某些语句，添加新代码段，提交；
* 切换到master分支，创建个Master类，修改Test类的同一部分语句，提交。

**合并过程**

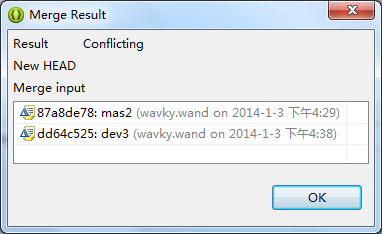
打开Git Repository Exploring视图：



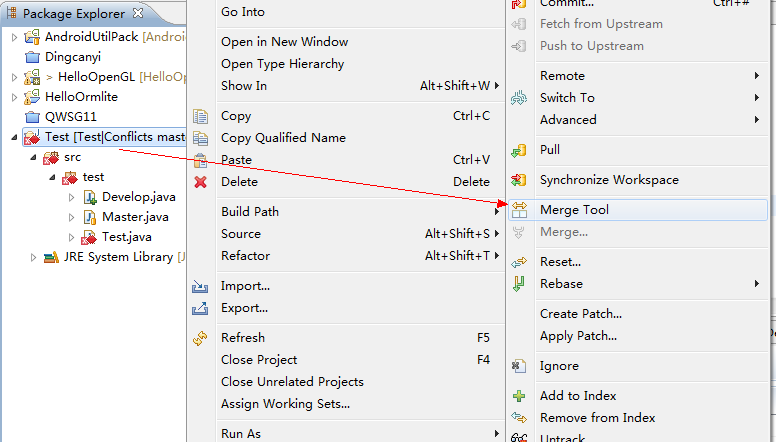
在Test库中，在develop分支上打开右键菜单，点击Merge：



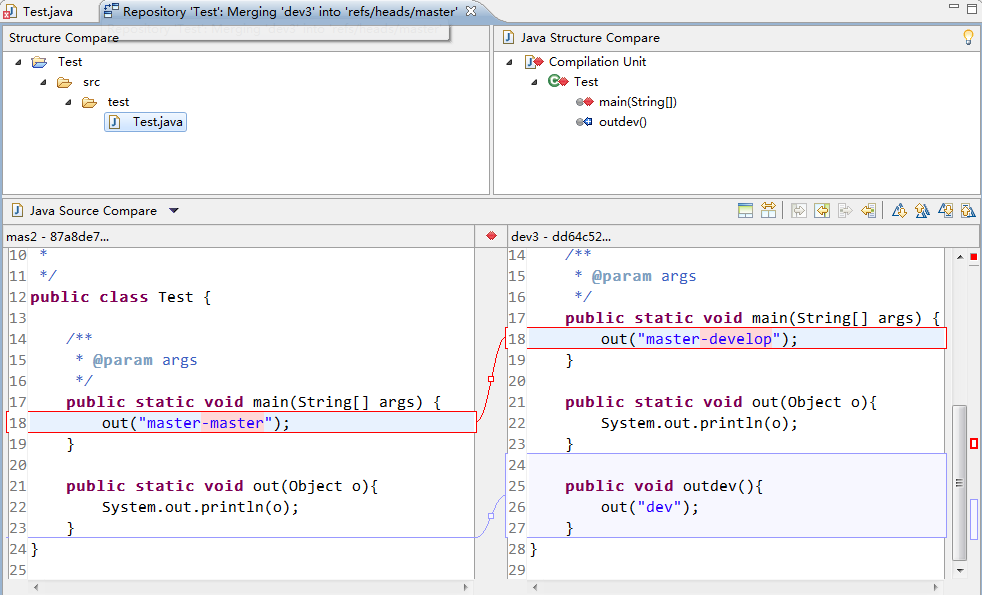
出现冲突提示框：



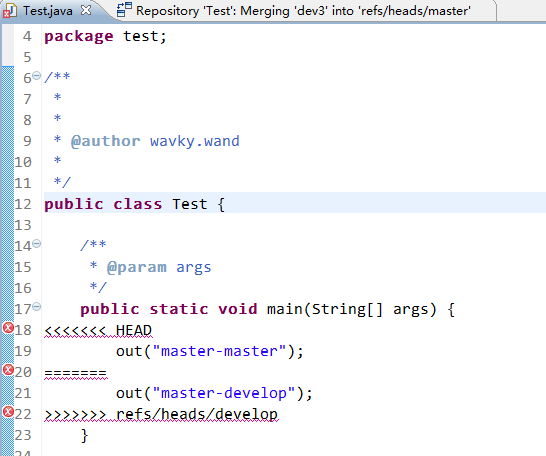
返回JAVA工程项目视图，可以看见Test工程已经自动进行了部分合并，添加了Develop类，但Test类存在冲突（有红色双箭头标记的文件才是冲突，茶色星星标记的文件没有实际冲突项），选中工程打开右键菜单，打开Merge Tool：



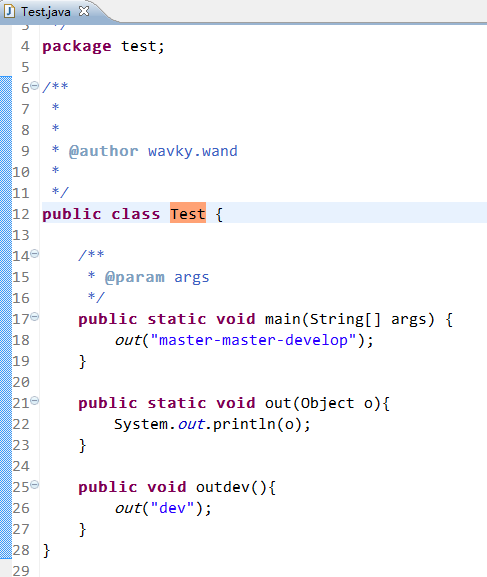
Merge Tool界面显示如下，左边为冲突文件的当前版本（master分支），右边为准备合并过来的目标版本（develop分支），手工把右边的代码copy到左边，或**至少随意更改下左边的文件**，保存。   注：右边的窗口有时候标题显示的版本节点不正确，可能是BUG，或者笔主理解能力不足所致。



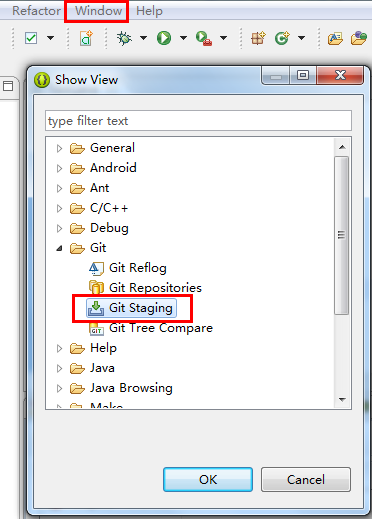
在Merge Tool中更改并保存后，当前冲突的Test类如下图中的各种神奇标记符号将自动消除：



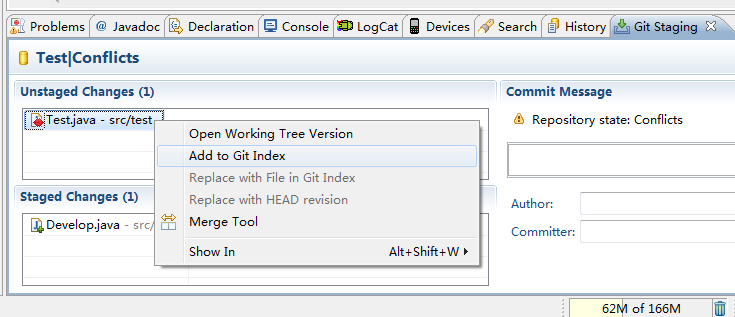
上图的红色神奇符号已经自动合并消除，并呈现类似下图的合并效果：



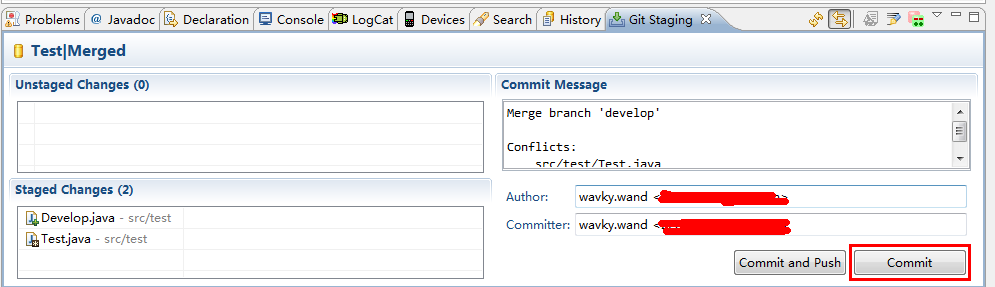
Window->Show View->Other，打开Git Staging视图：



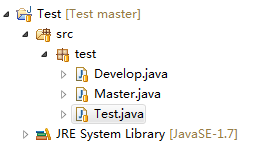
Git Staging视图中，Unstaged Changes部分包含冲突文件（包括手工合并前后），Staged Changes部分包含已经完成自动合并的文件，在已经合并完成的冲突文件Test.java上打开右键菜单，选中Add to Git Index：



Test.java被添加到Staged Changes下面，并自动生成Commit Message等内容，确认无误后点击Commit提交：



至此，Merge合并的冲突已经完全解决，从develop到master方向的合并已经完成：



若要从master合并到develop，只需切换到develop分支，（在Git Repository Exploring视图中）在master菜单上点击Merge即可。笔主使用的Egit2.2版本默认进行Fast-Forward方式的合并，最新的3.2版已经提供是否选择Fast-Forward的UI窗口。

\* 关于Egit3.2的**Fast-Forward合并**：提供该选择的UI窗口仅在JAVA视图窗口中，通过项目工程右键菜单中执行Team->Merge触发显示，通过上面使用git repository exploring视图简单调用Merge的方式仍然使用默认的Fast-Forward合并：

