cvs主干分支合并助手

部门：CDD-SWI

version 0.1

时间：2012-10-29

目录

[背景 3](#_Toc339297682)

[面临问题 3](#_Toc339297683)

[解决方案 4](#_Toc339297684)

[软件介绍 4](#_Toc339297685)

[工具的安装 4](#_Toc339297686)

[工具的配置 4](#_Toc339297687)

[工具的使用 5](#_Toc339297688)

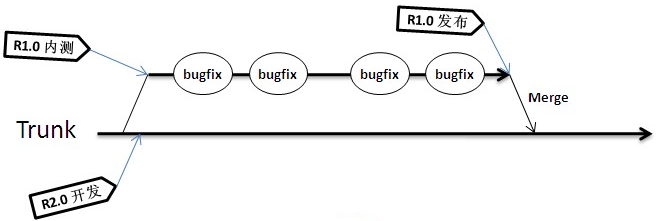
# 背景

目前团队使用cvs作为代码管理工具，因为项目特殊性，需要频繁的每隔1,2周就从主干上建新分支，在分支上每提交一个补丁或功能时都需要手工合并代码到主干，容易遗漏。而我开发的这个小工具就是为了比较方便的完成这件事情的。

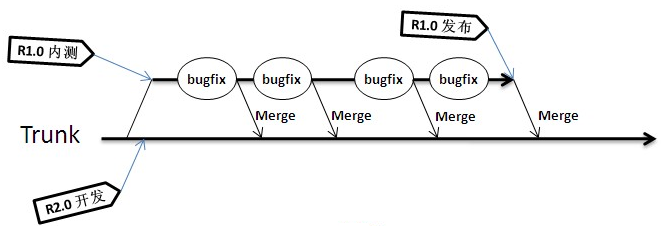
# 面临问题

目前我们团队的持续集成环境采用主干负责开发大版本，分支负责开发小版本的方式为现场提供版本。每次往现场发布一个大版本后，会为这个版本开一个新分支，当前版本在现场发现的bug会在分支上进行修正，而主干代码则负责开发下一期的新功能。而在项目开出新分支后，开发人员面临一个问题，就是如何方便准确地把分支上改动的bug补丁同步到主干上。以便下一个大版本里不会出现这个版本中的bug。

这里有两种方式，一是每个开发人员往分支上提交bug补丁后就不管了，当这次小版本生命周期结束时，要往现场发布大版本时，才统一把分支上的差异合并到主干上。这样好处是开发人员每次修改bug的速度会比较快，但是这样会导致最后合并时，首先是容易漏掉一些代码，其次是一次合并代码太多，容易出错。如下图所示：



二是每个开发人员往分支上提交bug补丁后，需要自己把代码合并到主干上。这样好处是每次改动的代码会及时同步到主干，代码量不太多，合并比较容易。但每次bug的提交会比较慢一些，对现场问题的响应速度没那么及时。如下图所示：



为了保证版本的稳定性，我们决定采用第二种方式。那如何提供开发人员合并的效率呢？之前我们的开发人员合并代码的方式一般都直接用eclipse的代码比较功能把本地主干和分支代码的差异比较出来，再一个个的合并，但这种方式会把所有的代码差异（包括自己的和别人的）都列出来，开发人员需要自己判别哪些是自己的，哪些是别人的。有的开发则会把这次分支修改过的代码记录在记事本中，但这种方式在代码改动少时没问题，代码多时容易导致遗漏。所以我们团队迫切需要一个功能，能够帮助我们在改动分支代码后，自动比较出自己修改过的代码，忽略别人修改过的代码，方便各个开发人员自己合并。

# 解决方案

为了比较完美解决以上问题，我设想了以下的补丁代码提交方式：

1 开发人员在分支上开发补丁代码。

2 本地测试分支补丁通过后，搜索出本地分支哪些代码被改动过。

3 把本地被改动过的代码和主干上相同代码文件做比较，合并。

4 提交分支改动的代码和主干合并后的代码。

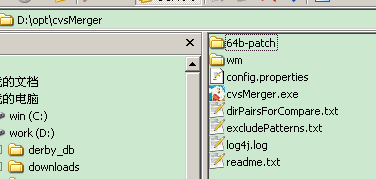
其中搜索和比较的操作可以由软件帮助完成，合并只能由人来做。代码提交仍然用eclipse提交。

# 软件介绍

软件是基于如下场景使用的。比如我在分支上修改一个功能，这个功能涉及到十几个源文件的修改。在提交分支代码到cvs前，先使用本工具执行本地主干和分支代码的比对，把本地分支改动的代码找出来和主干合并。合并后再把本地分支和主干修改过的代码都提交到cvs上。

## 工具的安装

工具是绿色软件，只要解压到某个文件夹即可



## 工具的配置

因为面向的是开发人员，且配置项不多，所以没有开发配置界面，需要手工修改配置文件

1 dirPairsForCompare.txt

这个文件是保存要比较的主干和分支的对应目录，比如主干代码在e:\trunk，分支代码在e:\branch，则添加一行。

e:\trunk|e:\branch

2 excludePatterns.txt

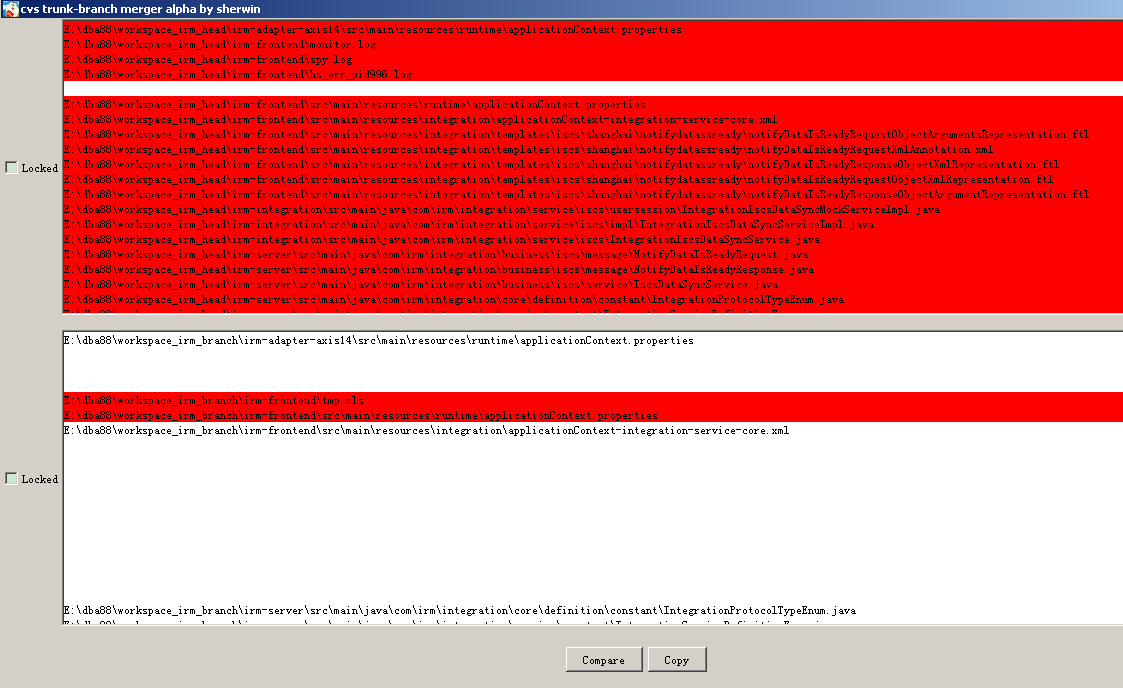
这个文件是保存要排除的文件列表，因为工程中有些文件是不需要合并的，比如日志文件，class文件等开发工具自动生成的临时文件。

## 工具的使用

刚打开工具时，点击比较按钮

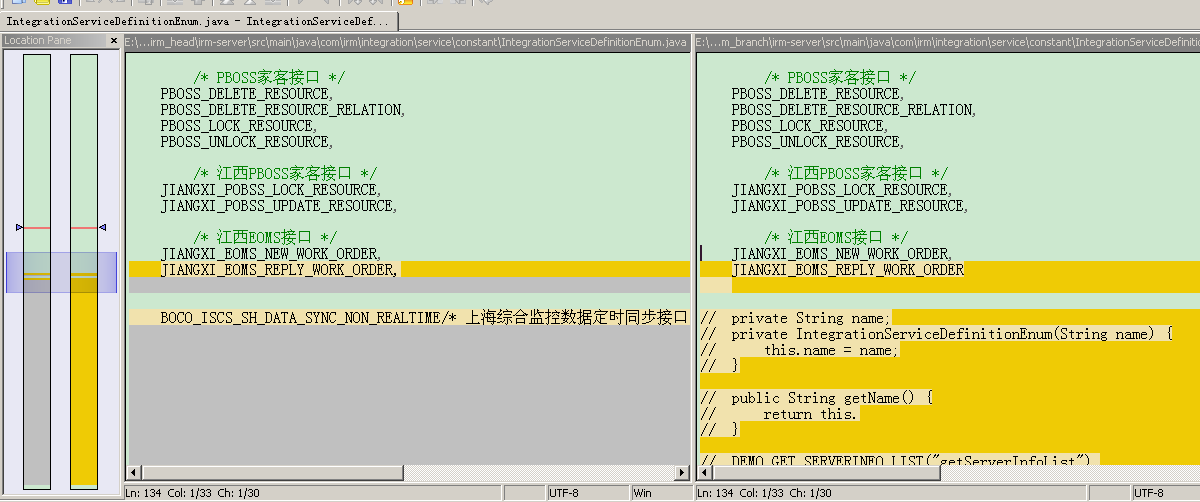


过一会就会显示出比较结果

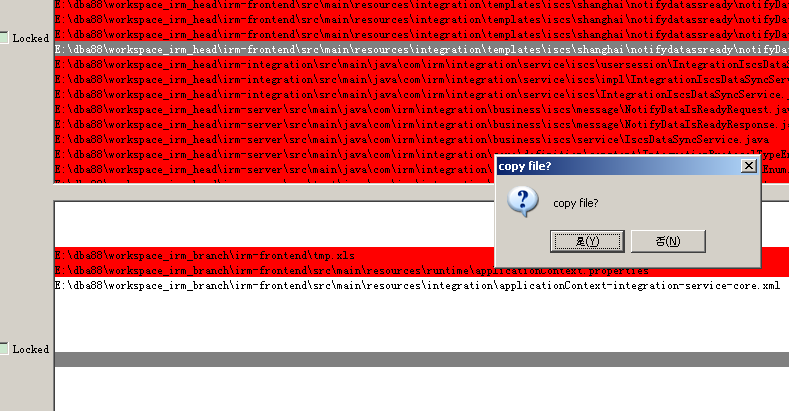


结果中列出主干和分支上本地修改过的文件，以红色标注。通过键盘上下移动可选择要合并的文件。

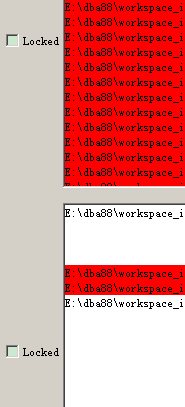




移到某项差异文件上按回车，就会打开比对窗口，这里可以进行合并工作，合并完毕按保存关闭。如果是按ctrl+回车，则可以打开文件所在文件夹，方便进一步操作。



如果某项差异是新增的文件，比如分支上新增一个文件，主干上没有，默认可支持复制操作，当按下回车时会提示是否复制文件。



左侧的Locked选项框是限制主干或分支的代码不能被修改，这是为了防止合并文件时误操作，比如本来是想合并分支代码到主干的，但可能不小心在操作过程中把主干的代码同步到分支上，加锁操作可避免这种情况。

为了进一步提高效率，以上所有操作都支持键盘快捷键。

合并完毕后在eclipse中提交即可。