第5章 Eclipse中CVS的使用

## 5.1 CVS简介

CVS是Concurrent Versions System（并发版本系统）的简称。它是一个开放源代码的项目，是当前最流行的版本控制系统，目前绝大部分Open Source项目都使用它来做版本管理。如果你还习惯用微软的VSS来进行Java项目的版本管理，那么用上Eclipse后，建议还是转为使用CVS。 CVS采用客户机/服务器体系，代码、文档的各种版本都存储在服务器端，开发者首先从服务器上获得一份复制到本机，然后在此基础上进行开发。开发者可随时将新代码提交给服务器，当然也可以通过更新操作获得最新的代码，保持与其他开发者的一致。 Eclipse本身内置了CVS客户端，只要再建立一个CVS服务器就可以使用这一功能强大的版本控制系统。CVS的功能虽强大，但一般项目通常只用到其20%的功能，所以只要了解最常用的操作就可以了，本章将以面向实际项目使用需要的方式来介绍CVS。

## 5.2 CVS服务器端的安装与配置

CVS起源于UNIX/Linux平台，关于UNIX/Linux平台下的安装使用CVS服务器端的文章，在网上多如牛毛，本节不再重复。CVS服务器在Windows平台的版本：cvsnt，它的安装有一定困难，本节将着重介绍cvsnt的安装。首先到cvsnt主页http://www.cvsnt.com/downloads/去下载它，下载方法参阅配书光盘的视频演示。目前的最新版本是cvsnt 2.0.58d，它可以安装在Windows NT/2000/XP/2003上。

（1）运行安装程序，不必更改它的任何默认设置，连续单击next按钮，即可完成安装。cvsnt默认安装在“C:\Program Files\cvsnt”目录下。

（2）通过Windows选择“开始→所有程序→CVSNT”选项，打开cvsnt的设置面板“Service control panel”，如图5.1所示，上面两个服务已经启动。 %注意：如果提示找不到rundll32.exe，则单击“浏览”按钮，指向“C:\windows\system32\ rundll32.exe”并运行它。

（3）单击图5.1中的Repositories选项卡，再单击Add按钮，创建一个存放版本文件的目录“c:/cvsfile”，如图5.2所示，单击OK按钮后在弹出的两个窗口中回答“是”。 %注意：建议在Name栏不要用自动生成的“/cvsfile”，而是改写成绝对路径“c:/cvsfile”，否则有可能在cvsnt安装后无法正常使用。 图5.1 Service control panel界面 图5.2 建立存放版本文件的目录

（4）转到“Compatibility”选项卡，将3个选项都选上，如图5.3所示。 %注意：这一步非常重要，否则CVS 2.0.58d无法在Eclipse 3.0.1下正常使用，如果你使用Eclipse 3.1M4版，则此步设置可跳过。 图5.3 Compatibility选项卡

（5）CVS客户端要连接上服务器，还需要在服务器端分配一个用户名，并且用户名要求必须是Windows的登录用户。本书登录Windows XP的用户名为giles，密码为空。进入Windows的命令行窗口，如图5.4所示，转到C:\Program Files\cvsnt目录下，输入命令cvs -d c:\cvsfile passwd -a giles，按回车键输入giles在CVS上的登录密码（本书设置密码为123456），此密码与XP可以不同。 %注意：安装好CVSNT之后，也可以通过“控制面板”里的“用户账号”来创建一个新用户，这时该用户自动成为CVS用户，密码相同。至此，CVS服务器部分的设置已经全部完毕，客户端已可以使用giles为账号来登录CVS。 图5.4 命令行窗口

（6）其他如果要将CVS的文件移植到另一台电脑上，或者原cvs服务器所在电脑重装了操作系统，这时就要涉及移植或恢复CVS。以本书配书光盘的cvsfile目录（后面章节的CVS备份）为例，读者只需要将此目录复制到本地电脑（复制后也可以改名），然后按照图5.2的操作将它注册一下即可。由此可知，备份CVS上的文件只需将cvsfile目录复制一份就行了，非常简单。如果读者对权限要求比较严格，那么就需要设置CVS用户的权限，CVS的权限是和操作系统权限相关的（用户都是操作系统的用户），也就是说读者需要设置Windows文件目录的权限。

5.3 CVS客户端的配置

5.3.1 配置Eclipse的客户端来连接上CVS服务器

打开Eclipse中的“CVS资源库研究”透视图，然后在左部的“CVS资源库”视图单击右键，在右键菜单中选择“新建→资源库位置”选项。在弹出的窗口输入CVS的相关信息，如图5.5所示，填完后单击“完成”按钮。 图5.5 添加CVS资源库 %注意：（1）资源库路径用CVS服务器上的绝对路径，即图5.2中Name文本框的值。（2）密码是上面用cvs命令行设置的123456。但如果是通过“控制面板”里的“用户账号”来创建用户的，那么此密码即为该用户的Windows登录密码。

5.3.2 将项目提交到CVS服务器上

将项目提交到CVS服务器上的步骤如下：

（1）+。右键单击项目名myproject，然后选择“小组→共享项目”选项，如图5.6所示。在弹出的“共享项目”窗口中，接受默认设置，直接单击“下一步”按钮。 图5.6 “共享项目”窗口

（2）如图5.7所示，窗口提示输入模块名，接受默认选项，单击“下一步”按钮。 图5.7 共享项目－输入模块名

（3）如图5.8所示，在窗口中选择要提交到CVS上的项目资源（即文件），不必作任何设置（默认为提交项目所有文件），直接单击“完成”按钮。 图5.8 共享项目－项目资源

（4）如图5.9所示，连续弹出两个窗口，都单击“是”按钮。又弹出一个窗口要求输入提交注释，写上一些说明文字（也可以为空），单击“确定”按钮，Eclipse开始将项目文件提交到CVS服务器上。 图5.9 “资源提交”对话框

5.3.3 CVS服务器上的项目导入到Eclipse中

如果新同事加入项目开发团队，这时他就需要将CVS服务器上的项目导入到自己的Eclipse中。为了方便模拟，再安装一个Eclipse，只要安装目录不同，两个Eclipse同时运行也不会有冲突。

将CVS服务器上的项目导入到Eclipse中的步骤如下：

（1）在主菜单选择“文件→导入”选项，弹出如图5.10所示的窗口。 图5.10 “导入”窗口

（2）选择“从CVS检出项目”，单击“下一步”按钮。弹出一个设定资源库的位置窗口，接受默认选项，直接单击“下一步”按钮，弹出如图5.11所示的窗口。 图5.11 填写CVS上的模块名称

（3）在模块名称文本框输入myproject，单击“下一步”按钮，弹出如图5.12所示的窗口。myproject这个名称不是项目名称，而是项目共享到CVS上时所取的模块名（模块名的设置参见图5.7），只不过通常将CVS模块名取成和项目名相同。 图5.12 “检出为”窗口（4）在项目名称文本框输入myproject（可任意取名）。Eclipse会以此名称生成一个项目，并将CVS的文件导出到此项目下。在该窗口中直接单击“完成”按钮，CVS服务器上的最新文件开始导入到本机中。 5.4 文件提交与更新的方法 5.4.1 CVS和VSS的不同之处对于用过微软VSS的读者，在介绍CVS文件操作之前，有必要了解一下两者的不同 之处：（1）用VSS必须check out（检出）文件才能修改此文件，而此时该文件将被锁定，其他人不能再修改（可以读），必须要等文件被上一个修改者check in（提交）才行，也就是说VSS同一文件一次只允许一个人修改。（2）CVS则使用完全不同的机制，多人可以同时修改同一文件。文件不会像VSS一样被锁定，所以也就没有修改文件之前要check out（检出）的操作，CVS的最基本操作是 “提交文件到CVS”和“从CVS更新文件到本机”。当然在CVS中如果多人同时修改了一个文件，在提交时会提示代码冲突，并提供一个窗口来合并这些修改，但这样的合并修改操作是很麻烦的，因此应尽可能地避免多人同时修改一个文件。 5.4.2 文件的提交和更新 1．将修改的文件提交到CVS 在HelloWorld.java中增加一条输出语句“System.out.println("第一次修改");”，然后保存。右键单击窗口右侧“导航器”视图中的HelloWorld.java文件，在弹出菜单中选择“小组→提交”选项，如图5.13所示。 图5.13 提交界面 2．从CVS更新文件到本机删除掉HelloWorld.java中的语句“System.out.println("第一次修改");”然后保存，这时要将CVS服务器上的最新代码更新到本地，有以下两种方法：

（1）右键单击HelloWorld.java文件，在弹出的菜单中选择“替换为→HEAD的最新内容”选项。

（2）右键单击HelloWorld.java文件，在弹出的菜单中选择“小组→与资源库同步”选项，在弹出的同步界面，如图5.14所示，单击“将当前更改从右边复制到左边”按钮，然后保存。 图5.14 资源库同步界面读者可能会问：“右键单击HelloWorld.java文件，然后选择‘小组→更新’这种更新方法，为什么不用呢？” 这种更新方法是将CVS的最新版更新到本机，当本机文件没有修改的时候才用这种方法。如果对HelloWorld作了修改，CVS上的反而是旧内容了，这时再用这种更新方法则对本地的HelloWorld.java文件不会产生任何影响。 5.4.3 解决文件提交的冲突在多人项目开发中，由于同时修改一个文件，一时就会产生提交冲突，本小节就来说明一下如何解决这种冲突。在上面检出项目时，已经另外安装了一个Eclipse，相当于多了一个开发者，原来的称为Giles，新加入的叫Tom。Giles和Tom电脑上的HelloWorld.java的版本都是1.2。这时Giles在文件中增加一条语句“System.out.println("Giles第1次修改");”，同样Tom也在文件中作了类似修改，如图5.15所示。 图5.15 两开发者对代码的修改情况 Giles先提交HelloWorld.java文件，这时版本变成了1.3，由于Giles是第一个提交的，所以他不会遇到代码冲突。Tom编程速度慢一些，所示后提交，这时就会和Giles先提交上去的代码产生冲突。因为如果用Tom的代码更新CVS，则Giles的修改就会被冲洗掉，所以Eclipse此时就会报文件冲突，不允许直接提交。那么Tom该如何提交自己的修改呢？解决办法如下：右键单击“HelloWorld.java→小组→更新”选项，如图5.16所示。 图5.16 更新后的界面此时HelloWorld.java的版本已经变成了1.3，更新后的冲突也已标记出来。这时就可以根据代码中的标记来合并修改，修改完毕后再提交到CVS，此时版本变成了1.4，如图5.17所示。 图5.17 代码合并修改后的界面 Eclipse考虑得很周到，在文件更新时，会将Tom的旧文件保存为一个称为“.#HelloWorld.java.1.2”的备份。这个备份文件不会被提交到CVS，因为Eclipse中已经将这类文件作了忽略设置。忽略设置的界面如图5.18所示，在主菜单选择“窗口→首选项”选项可打开此窗口。 5.4.4 如何忽略掉不想提交的文件假设当提交一个项目时出现如图5.19所示的情况，图中的bin目录包含的是编译后的class文件，没有必要将它提交到CVS上。我们可以右键单击它，然后在弹出的快捷菜单中选择“添加至.cvsignore”，这样bin目录就被忽略掉，不会提交到CVS上。 图5.18　提交到CVS时被忽略的资源的设置界面 图5.19 忽略提交 5.4.5 实践建议

（1）把项目做成模块目录（Java包），各人负责自己的模块和目录，这样就不容易造成重复修改。

（2）做到每日提交，即每天下班之前都要将各自完成的代码提交到CVS上，并且提交的代码要求是完整可运行的代码，关键是要保证其他人更新你的代码之后不会引起错误。

（3）对于共用型的文件，要做到即时修改，即时提交。

（4）任何人在修改文件之前，先更新CVS的最新版到本机。

（5）项目团队的Eclipse等开发环境（安装目录、软件版本等）要保持一致。这样就可以将整个项目共享到CVS，团队成员也可以从CVS上更新整个项目，而不必担心因为一些项目配置文件不同导致更新项目后出错。另外，当新团队成员加入时，只要导入CVS上的整个项目，无须再配置项目，就可以即刻开发运行。

5.5 在CVS上为软件打包一个版本在开发告一段落时，就要打包一个版本，打包版本的操作步骤如下：

（1）右键单击“项目名”，在弹出的快捷菜单中选择“小组→标记为版本”选项。此时如果有修改没有提交到CVS上，则会弹出下如图5.20所示的一个提示框，如果单击“确定”按钮，则未提交的修改就不会包含在本次打包的这一版本中。

（2）如图5.21所示，在窗口中为此版本起一个名称为V001。由于此名称可能已存在于CVS中，所以下面还有一个复选框，如果选上该复选框，在打包时就会替换掉同名的老版本；如果没有选上，则在有同名版本时，就会弹出错误提示框。 图5.20 确认未提交的更改 图5.21 给版本起名如果版本已经打包，但后来发现有一个文件有点问题，于是想把这一个修改的文件更新到上一个版本中，但另外一些修改的文件则不想更新到上一个版本。这时如果再通过右键单击项目名来打包，则项目所有修改的文件都会被更新到上一个版本中。正确的操作步骤如下：

（1）将修改的文件提交到CVS上。

（2）右键单击该修改文件的“文件名”，在弹出的快捷菜单中选择“小组→标记为版本”选项。

（3）在图5.21中，还是起名V001，并且要选择复选项，然后单击“确定”按钮。此操作和打包项目版本惟一不同之处是：前者右键单击“项目名”，后者右键单击修改文件的“文件名”。 %说明：CVS打包版本并非是将项目文件再复制一份，如果去c:\cvsfile目录中查看提交的项目文件，就会发现项目文件只有一份，这一份文件中会记录着每一次的更改，所以CVS打包版本只是做了一个更改位置的标志。

5.6 将项目替换成CVS上的其他版本接着5.5节内容，当打包成版本之后，如果要将此版本还原回项目中，则操作步骤如下：右键单击“项目名”，然后在弹出的快捷菜单中选择“替换为→另一个版本或分支”选项，得到如图5.22所示窗口，选择版本V001，单击“确定”按钮。 %注意：此操作会将本地所有的修改都覆盖掉，所以在操作之前先把本地修改提交到CVS上保存。 图5.22 CVS的分支版本列表

5.7 修改旧版本的BUG 如果已经发布了软件的1.0版，而现在正在开发2.0版，但用户在使用1.0版的时候发现了一些BUG，这时就需要修改1.0版的这些BUG，具体修改操作步骤如下：

（1）将CVS导出成另外一个项目。参见5.3.3小节，其他步骤都一样，仅在最后一步时，输入另外一个项目名myproject2。这样分成两个项目后，修改旧版本和开发新版本就可以同时进行而不互相干扰。

（2）用5.6节的替换版本操作将myproject2项目替换成V001版本（假设V001就是出问题的1.0版）。如图5.23所示，在图中可以看到myproject2的后面跟着一个V001版本 标志。 图5.23 从CVS导入的myproject2

（3）这时还不能在myproject2上修改，即使修改了也提交不上CVS。因为还差一步，即需要为V001版创建一个分支，操作步骤如下：右键单击myproject2的项目名，然后选择“小组→分支”选项，在弹出如图5.24所示的窗口中输入分支名V001\_patch，并单击“确定”按钮。 %说明：很容易把“分支”和“版本”看成类似的概念，但实际上“分支”是和HEAD的概念比较接近，“提交”操作只能将文件提交到HEAD或分支上。

（4）如图5.25所示，myproject2的项目名后就有了一个分支名V001\_patch，表示该项目已经在这个分支下工作了。 图5.24 设定分支名 图5.25 myproject2在V001\_patch分支下

（5）现在可以在myproject2项目中将BUG修改好，并提交到CVS上了。不过要说明的是，修改不是提交到HEAD上，而是提交到分支V001\_patch上，所以也就不会影响到软件2.0版的开发。提交之后，文件尾部的版本号也起了变化，如图5.26所示。 图5.26 在分支下的程序版本号

（6）当修改完BUG后，就可以给软件再打一个版本V100\_1，并将修正了BUG后的版本发布给用户。

5.8 本 章 小 结本章介绍CVS的从服务器安装到客户端具体操作的整条开发线的使用流程，其中文件提交与更新是项目开发中使用最频繁的操作，本章详细讲述了具体的操作流程，并给出了由项目开发经验中总结出来的实践建议。通过本章的学习，读者已可以轻松地在项目中使用CVS来进行版本管理。

发表于 @ 2008年07月07日　14:51:00 | 评论( 0 ) | 编辑| 举报| 收藏

新一篇:Java文件操作

本文来自CSDN博客，转载请标明出处：http://blog.csdn.net/sucrejee/archive/2008/07/07/2621413.aspx