**[ubuntu搭建cvs服务器](http://suixinyuano0.iteye.com/blog/616205)**

卸载用命令：

apt-get remove cvsd;

apt-get remove cvs;

### 分享ubuntu之CVS

第一步、安装CVS  
安装cvs的资料很多，但是并不是所有的资料都是通过实践检验的，也不是所有的资料都是适合初学者的，像我这样的纯新人，拿了资料也不一定能找到正确答案，所以，就像鲁迅先生所讲的：地上本没有路，走的路多了，也就有了路！开始犯错！   
  
安装CVS有一下几个问题，希望大家能参与进来讨论一下：   
  
Q1：安装cvs服务器时，是否需要用root帐户？或者切换到root帐户权限下操作？   
  
A1：这个算不上答案，只能算我的经验。事实上，刚开始我是在普通用户下安装cvs的，不过安装到初始化repos时，就不好使了，往下的步骤无法执行。   
发现这个问题，是因为我发现终端的提示符是$,而资料里面是#，后来就查到资料说要进入root权限，于是我又查到如何进入root，现在开来也很简单， 在$后输入 sudo passwd（如果你是第一次进入的话），然后就可以设置root帐户的密码，以后再进入root是，只要在$后输入su就可以了。   
切换到root权限下，再按照网上的资料进行操作，就一切ok了！   
虽然这是实践的结果，但是我还是怀疑，是否只有在root权限下才可以按装cvs呢？  
  
最近开始写一些程序，发现自己经常把程序改得很乱，以至后来自己都看不明白了，为解决这个问题，我在ubuntu下配上了cvs,这是一个非常优秀的版本管理工具，如下是我参考网上同胞们的文章后的一个个人总结.

1.联网，配好源，执行:shell>sudo apt-get install cvs

2.shell>sudo apt-get install cvsd 。此时会问你repository的位置，我按默认配制，没做修改，建议你也不要改，如果你不清楚的话。（）

3. 创建这个/myrepos。  
shell>cd /var/lib/cvsd  
shell>sudo mkdir myrepos

4.初始化这个/myrepos。  
shell>sudo cvs -d /var/lib/cvsd/myrepos init

5.设置cvsd的用户认证方式，缺省的是系统级的，一般来说cvs的用户都是单独设置的。  
shell>sudo vim /var/lib/cvsd/myrepos/CVSROOT/config  
添加以下几行：  
SystemAuth=no  
PamAuth=no  
LockDir=/tmp/myrepos  
然后保存退出。注意此处的/tmp/myrepos是指相对于RootJail的路径(此条参考网上所写，对此我暂时还不清楚）。（看了/etc/cvsd/cvsd.conf里面确实有个RootJail /var/lib/cvsd，估计上面的路径都是和这个有关）

6.创建LockDir  
shell>sudo mkdir /var/lib/cvsd/tmp/myrepos

7.添加用户  
#sudo cvsd-passwd /var/lib/cvsd/myrepos chenheng #你想建的用户代替"chenheng"

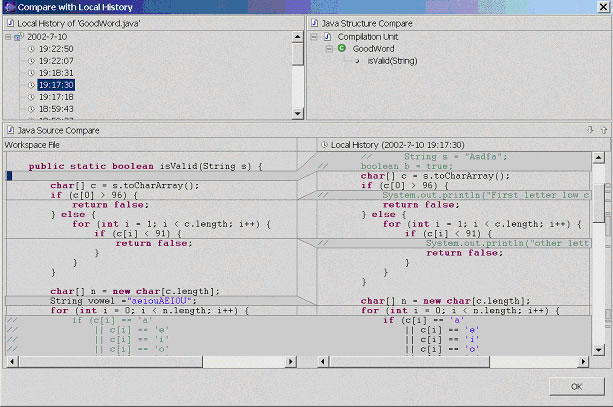
8.cvsd默认端口是2401,你可以改别的端口  
#sudo vim /etc/cvsd/cvsd.conf  
Listen \* 2401  
此外缺省有：  
RootJail /var/lib/cvsd  
Repos /myrepos  
注意此处的Repos的路径是相对于RootJail的,前面安装cvsd时默认也是这样的，你不需要做修改。  
  
10.shell>sudo chown cvsd:cvsd /var/lib/cvsd -R      让cvsd运行这个服务，这个用户在安装cvsd时已经建好。

11. 重新启动cvsd。  
shell>sudo /etc/init.d/cvsd restart

12. 在eclipse使用此cvs,选中项目名称，右键选team->share project,以下是客户端的一些参数，  
connection type:pserver  
host:127.0.0.1      当然我这里是本地主机做为服务器同，你也可以在远程主机设置  
port:2401  
repository path:/myrepos  
user:chenheng  
password:\*\*\*\*  
以上转自：http://blog.chinaunix.net/u/30752/showart.php?id=321662，经在ubuntu8.04下测试通过  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
这个转自:http://blog.csdn.net/taina2008/archive/2007/08/03/1723679.aspx

Eclipse的内置版本支持

Eclipse本身内置了一定程度的版本支持，也就是所谓的Local History。在Eclipse中编辑的文件每次存盘都会留下记录，可以随时与历史记录比较，恢复到某一个时刻的状态。在资源视图（Resource Perspective）或者Java视图（Java Perspective）中的文件上点右键，选择Compare With，Local History，就会得到如下的画面，可以很清晰看到各个版本的区别。

**图1：Eclipse中的内置版本支持：Local History。**  


如果需要，则可以通过Replace With，Local History来恢复到任一个以前的版本。当然，这样版本的历史是有限制的，可以在Eclipse的Preference中设置。在 Preference D>Workbench－>Local History中，可以设置保持最多保持几天的文件，每一文件的版本数目，历史文件的大小限制。

但是，这样的版本系统只实现了最基本的版本功能，如果想把文件某一个状态标记为一个版本，加上注释（仅仅从Local History中的保存时间很难看出每一个版本的特点，也难以找到重要的关键版本），或者想把某个目录乃至整个项目版本化，Eclipse内置的 Local History都是无能为力的。

好在Eclipse基于开放的思想，采用了业界标准的版本控制系统CVS，这样不仅很好地实现了版本控制的功能，对于已经熟悉CVS的开发者来说也 更容易使用，更重要的是，可以为团队开发提供更加灵活和开放的选择：项目的数据可以存储在一个CVS服务器内，不同的开发者甚至可以采用不同的开发工具， 只要这些工具都支持CVS。也许这时我们可以理解为什么Eclipse放弃了Visual Age Java的业已很成功的ENVY系统，这一点很好地体现了Eclipse的开放性和对其他软件的良好互操作性。

CVS

CVS是Concurrent Versions System（并发版本系统）的缩写，基于Unix体系中成熟的SCCS（Source Code Control System）和RCS（Revision Control System）开发，是一个开放源码的项目，目前已是版本控制系统的主流软件。一个很常见的使用CVS的场合，就是开放源码项目。由于开放源码项目的开发 者的分布性，对于版本管理的要求更加严格，而目前大部分的开放源码项目几乎都是采用CVS来管理源代码，CVS的标准性和强大可见一斑。

CVS采用客户机／服务器体系，代码以及各种版本存储在中心服务器内，每一个个体开发者开发时都首先从服务器上获得一份自己的拷贝，在此基础上进行 开发，以避免直接影响服务器上的数据。开发者可以随时把自己的新代码提交给服务器，并通过更新获得代码的最新状态，保持与其他开发者的一致。

CVS对于网络是透明的，开发者可以使用客户端软件（几乎所有的平台上都有相应的客户端软件）在任何时候，任何地点通过网络来获取最新的代码。

对于Eclipse的开发者而言，Eclipse本身内置了CVS支持，不需要使用其他客户端软件，只要建立一个CVS服务器，就可以使用这一强大的版本控制系统了。

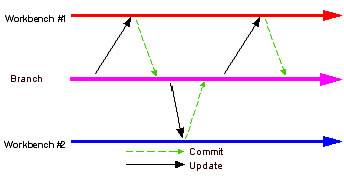
1.安装CVS，在新立得包管理器里面选中cvs（系统-系统管理-新立得软件包管理器，在右边的软件包列表中随便选中一个，然后键盘上按cvs这 三个字母，就会定位到以这三个字母开头的软件包），然后点击鼠标右键，在菜单中选中标记以便安装。然后点击上方工具栏中的应用，系统会自动下载包，然后安 装。   
  
2.安装CVSD，同上，选中cvsd，安装。安装cvsd后，会提示设置repository的位置，这个地方只要输入相对于 RootJail的位置就可以。RootJail在/etc/cvsd/cvsd.conf中进行设置，缺省的RootJail是/var/lib /cvsd。也就是说，在这一步里面输入/myrepos就可以了。   
  
3. 打开一个命令行窗口，创建这个/myrepos。   
#cd /var/lib/cvsd   
#sudo mkdir myrepos   
  
4.初始化这个/myrepos。   
#sudo cvs -d /var/lib/cvsd/myrepos init   
  
5.设置cvsd的用户认证方式，缺省的是系统级的，一般来说cvs的用户都是单独设置的。   
#sudo gedit /var/lib/cvsd/myrepos/CVSROOT/config   
添加以下几行：   
SystemAuth=no   
PamAuth=no   
LockDir=/tmp/myrepos   
然后保存退出。注意此处的/tmp/myrepos是指相对于RootJail的路径。   
  
6.创建LockDir   
#sudo mkdir /var/lib/cvsd/tmp/myrepos   
  
7.添加用户   
#sudo cvsd-passwd /var/lib/cvsd/myrepos cvsuser   
  
8.编辑cvsd的配置文件，配置那个该死的RootJail。   
#sudo gedit /etc/cvsd/cvsd.conf   
缺省的应该是：   
RootJail /var/lib/cvsd   
Repos /myrepos   
注意此处的Repos的路径也是相对于RootJail的。   
另外，如果有些变态的网管喜欢封端口的话，这里可以设置cvsd的监听端口，比如可以改缺省的2401端口为8080。   
Listen \* 8080   
  
10.设置目录权限。   
#sudo chown cvsd:cvsd /var/lib/cvsd -R   
  
11. 重新启动cvsd。   
#sudo /etc/init.d/cvsd restart   
  
12. 在eclipse或者其他的cvs客户端中配置使用此cvs服务器。   
connection type:pserver   
host:your ip   
port:8080   
repository path:/myrepos   
user:cvsuser   
  
其实这个cvsd的安装，主要是参照它的README，只不过它是一个压缩文件，一开始没想解开看看而已。/usr/share/doc/cvsd/README.gz。

在Eclipse中使用CVS系统

前面已经提到，CVS的数据存放在服务器的数据库中。为了支持Concurrent这一并发特点，CVS使用了一个分枝（Branch）模型，以保 证不同开发者的相对独立，但是又高度集成。分枝可以看成一个开发团队共享的工作区（Workspace），在CVS数据库中，有一个特殊的分枝称为 HEAD，代表主要工作流。

开发者可以提交（Commit）自己的工作，并通过更新（Update）与其他人的最新修改保持同步。提交是把自己的修改提交给数据库，称为输出 （Outgoing），更新则是得到其他人的修改，称为输入（Incoming）。每次提交之前，都应该先更新，以保持与最新状态同步。

因此，在Eclipse中使用CVS进行团队开发，理想的开发流程应该是这样的（具体操作方法随后介绍）：

**图2：团队开发流程**  


1. 从最新状态开始。开始工作前，要保证所有资源与最新的分枝状态一致。对于从头开始的新项目，首先要连接服务器，设定存储那些资 源文件。对于在服务器数据库中已经有记录的项目，首先要通过更新来保证资源状态最新。如果本地的工作没有需要特别保存的，可以直接把数据库中的最新版本 Check Out As Project，或者Replace With，Latest from Repository。
2. 进行本地工作，保存修改。
3. 同步。当做好提交工作的准备后，要和服务器数据库同步。
   1. 首先应该更新（Update），检查所有的Incoming改变，输入到本地，看是否会影响自己的工作，是否会造成冲突，破坏完整性等等。
   2. 检查完更新后，可以确认自己的工作和最新的Branch内容是一致的，就可以提交（Commit）自己的修改了，标上适当的注释。

下面介绍这一开发流程所用到的主要操作。

对于一个新的项目而言，首先需要做的就是设置Eclipse来连接CVS服务器。首次连接将与服务器同步（Synchronize）。

首先在Eclipse的资源视图（Resource Perspective）或者Java视图（Java Perspective）中的项目上点右键，选择Team，Share Project，将出现提示窗口，选择是使用已知的CVS数据库位置还是新建数据库位置。对于第一次连接CVS服务器的情况，将需要首先定义连接所需的服 务器参数和路径，用户名，密码等等。这些参数将得到保存，以后如果需要重新连接，就可以直接使用已知的CVS数据库设置。

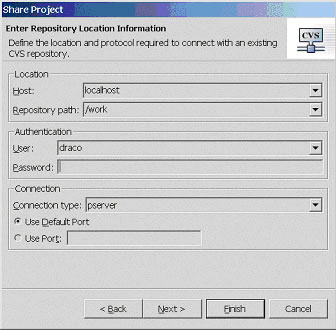
这里我们选择新建一个数据库位置，进入下一个输入画面。

这里的主机名（Host）可以输入localhost（在Windows NT 4下不可以这样输入），因为我们的CVS服务器就在本机上。

接下来输入/work，设定我们的项目数据存放在这一数据库根路径下。

然后输入用户名和密码。

连接类型选择pserver，端口是缺省的不用改动。当然，如果你修改过CVS服务器使用的端口，这里就要保持一致。

**图3：在Eclipse中设置使用CVS系统**  


接下来可以直接点击Finish完成设置。如果点击下一步，可以继续定义在CVS中是否使用与项目名不同的名字作为模块名，缺省是使用相同的名字。

现在我们完成了项目与CVS服务器的连接，项目已经和CVS服务器同步了。

但是这并不意味着CVS已经开始记录版本信息。因为在一个项目中可能有相当一部分资源文件不适合存储到数据库中记录版本信息，比如编辑器生成的临时 文件，编译生成的.class文件，或者某些编译过程生成的二进制文件等等。因此，Eclipse并不自动把所有文件存储到数据库中，需要开发者手工指定 把那些文件存储到数据库中，也就是加入版本控制add to version control。对于单个文件的指定，可以随时在资源视图（Resource Perspective）或者Java视图（Java Perspective）中的文件上使用右键，选择Team，Add to Version Control。如果需要存储整个项目或者同时指定多个文件，则要使用同步（Synchronize）视图。

注意项目与CVS服务器连接完成并同步后，在Eclipse右下角的Tasks和Console的位置新出现了一个Synchronize窗口，下 方状态栏也显示了这次同步的状态信息。双击Synchronize窗口的标题栏使之最大化，我们可以看到窗口上半部分列出了所有尚未指定加入数据库的资源 和文件。在这里我们可以同时选择多个文件，甚至选择整个项目，使用右键，选择Team，Add to Version Control，就可以把多个文件或者整个项目加入到数据库中，开始记录其版本信息。

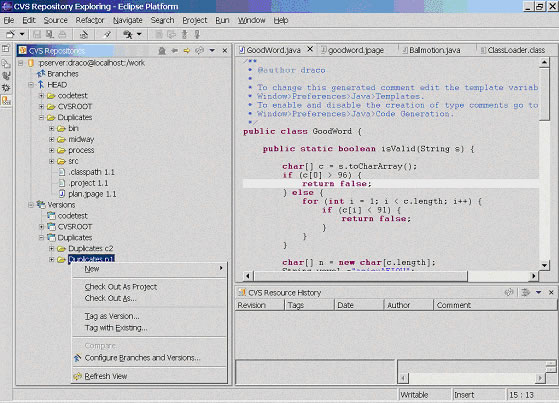
我们可以看到，这时的同步视图上标明了处于Outgoing模式。同步（Synchronize）视图有Incoming模式和Outgoing模 式，对应于提交（Commit）和更新（Update）操作，表示目前修改是来自本地还是来自服务器。当然，视图也可以同时显示Incoming和 Outgoing方向的变化。当有变化时，在下半部分的比较窗口可以显示不同版本之间的差异和变化。进一步地，可以通过工具条上的合并（Merge）操作 来消除对同一文件不同修改所带来的冲突。

在此之后，当完成对文件或者项目的修改，需要确认修改，把最新的状态存储到数据库中去，就需要执行提交（commit）操作。类似的，我们可以在资 源视图（Resource Perspective）或者Java视图（Java Perspective）中的文件上使用右键，选择Team，Commit；或者在同步（Synchronize）视图中同时提交多个文件乃至整个项目。 注意，每次提交时都会提示输入注释，这是很重要的，作为每一个状态的说明和提示。

当我们需要把文件或者项目的不同版本进行比较或者替换时，可以在资源视图（Resource Perspective）或者Java视图（Java Perspective）中的文件上使用右键，选择Team，Compare With或者Team，Replace With来完成。进行比较时将打开一个比较编辑器，很好地表示不同版本之间的差异，和Local History类似。

如果要了解CVS数据库的内容情况，可以使用CVS数据库浏览视图（CVS Repository Exploring Perspective）。我们可以点击Eclipse左上角的开启视图（Open Perspective）按钮，增加这一视图到左边的视图浏览栏里。前面已经提到，HEAD代表CVS数据库里的主流内容。展开Versions，可以看 到以往的版本。另外，在右下角有数据库的历史记录标明每次修改的相关信息。

如果需要导出某一版本，就要用到Check Out操作。在某一版本上右键选择Check Out As Project，这将把这一版本导出为同名的项目，从而可以覆盖已有的状态，或者选择Check Out对话框来导出为其他项目。

**图4：CVS数据库浏览视图**  


安装版本信息：  
12.04 server  
  
这是开头的一些废话： CVS在版本控制里还是不错的，C/S结构比较适合在公司开发代码使用。  
其实CVS还是比较好配置的，只要你用心，就没有问题。  
按着我的步骤做，你就能无错配置CVSD服务器。  
网上的教程很多，但说真的都是不断的随意没节操的COPY,根本没有验证过是否OK。  
好了，吐槽完了。下边进入正题，这篇教程是很基础的，但也需要你懂的一些vim和系统的基本操作。  
  
  
//-----------------------------------------------------------------------------------------------  
sudo apt-get install cvs //安装CVS的客户端，方便配置成功后调试  
  
sudo apt-get install cvsd //安装CVS的服务端  
我的安装后出现了一堆乱码，我直接回车了，这个不影响，因为后边可以配置的。  
这里的配置主要是让你决定代码仓库名称的  
  
//-----------------------------------------------------------------------------------------------  
现在，我想把我的CVS 根目录放到/home/name/cvsd下。  
但是安装后默认的根目录在/var/lib/cvsd.  
  
执行：  
cd /home/name //进入name目录  
sudo mkdir cvsd //创建cvsd文件夹  
sudo cvsd-buildroot /home/name/cvsd //把默认的CVS根目录设置为/home/name/cvsd  
//------------------------------------------------------------------------------------------------  
现在，我们的根目录好了，那么应该有一个根目录下的子目录，也就是代码仓库的项目文件夹。  
项目文件夹可以设置多个。  
  
sudo vim /etc/cvsd/cvsd.conf  
会看到这个 RootJail /home/name/cvsd ,这行其实就是上一步修改根目录配置的，在这里也可以修改的。  
我们要修改的是这一行： Repos /mytest ,这行就是定义项目文件夹的，可以设置多个。  
这里/mytest 是指根目录下的 mytest文件夹，这个文件夹还没有建立，所以我们设置好后就要创建了。  
cd /home/name/cvsd  
sudo mkdir mytest  
如果需要设置CVS服务器的绑定端口的话，可以把 Listen \* 2401 该为自己想要的。  
比如： Listen 192.168.1.1 8888 就是绑定在本地IP地址的8888端口上。一般该成你的IP和端口  
//------------------------------------------------------------------------------------------------  
现在我们要设置权限的问题，因为我们自己建立的文件夹是用sudo 建立的  
而现在需要设置为cvsd这个安装后自动建立的用户和组。  
  
sudo chown -R cvsd:cvsd /home/name/cvsd   
//------------------------------------------------------------------------------------------------  
现在需要初始化项目文件仓库:也就是我们的根目录下的项目mytest。  
  
sudo cvs -d /home/name/cvsd/mytest init  
  
执行无错后会在mytest下生成CVSROOT文件，这个就是控制mytest的一些配置文件了。  
//------------------------------------------------------------------------------------------------  
现在来配置有关mytest用户的设置  
sudo chmod o+w /home/name/cvsd/mytest/CVSROOT/config //加上写权限，默认没有，不能修改  
sudo vim /home/name/cvsd/mytest/CVSROOT/config  
  
找到 #SystemAuto=no 改成 SystemAuto=no //其实就是启用用户和密码验  
  
sudo chmod o-w /home/name/cvsd/mytest/CVSROOT/config //去掉写权限，不然会出错  
//------------------------------------------------------------------------------------------------  
上一步只是设置验证方式，下来需要添加用户和密码了。  
  
sudo cvsd-passwd /home/name/cvsd/mytest/ user-name(你自己的用户名)  
提示输入两次密码。  
  
//------------------------------------------------------------------------------------------------  
测试下我们的成果吧：  
sudo /etc/init.d/cvsd restart //重启CVSD服务器  
cvs -d :pserver:user-name(你的用户名)@192.168.1.1:8888/mytest login //登录  
cvs -d :pserver:user-name(你的用户名)@192.168.1.1:8888/mytest checkout //检出代码  
  
//基本过程就这些了，如果你有什么问题，可以加我QQ：496393411