CVS: The Concurrent Versions System

LATEST AND GREATEST

If you're interested in helping to develop Open Source software then getting to know the Concurrent Versions System is the logical first step. Colin Murphy shows us how to get started with CVS

Software development under Linux can be a real joy, thanks to the free movement of source code. Anyone can get involved in moving a project forward and you may not even need to have all that much programming experience to make a difference. In some cases you may not even have to code at all to get involved; what some projects desperately need are graphics designers and people to help write documentation.

To get involved, you first need access to the project and for that you need access to the source code. Part of the beauty of Open Source programming is the opportunity it gives you to look at other people's work and, if you feel up to it, to change it, possibly even to improve on it. What is even more useful is the chance to see the additions that other people have made to a code base for a project, to see how it grows and changes over time.

Software bugs

Developing software and developing bugs in that software go hand in hand, and bugs are bound to appear in software development. When projects are open to a wide-ranging 'team' of developers, the danger that bugs will be introduced is increased. More importantly, the nature of these bugs may not even be noticed until much further along the project's development cycle. Such bugs may not even be caused by bad or plain wrong programming – it's more likely that a previously written piece of code proves not to tie up well enough to the project now.

So how do program developers get around this major problem? What is needed is to keep copies of all of the old versions of the program development so that you have something to refer back to if and when something appears to be going wrong.

Seeing as though a change to a piece of code correcting an error might be only a few bytes long, keeping a back-up of all previous pieces of code would result in a huge archive of very similar data. What is needed is a system that records just the changes made to the code base. Thankfully, that's why we have CVS, or the Concurrent Version System, which maintains a central repository for people to access and modify.

If you're only just starting out in program development you may only have come across CVS very briefly, when exploring the very latest version of a piece of software, that cures a bug or adds a much-needed feature that's just not present in the stable version. With that must come a warning: you must always remember that this is an unstable version and to treat it as such. By using it, you should also remember your contribution to the project – bug reports.

For the most part you will be using CVS in a Server/Client mode, most likely via the Internet. CVS is a command line utility, though there are some graphical front-ends, which we will round the article up with in a moment.

Firstly, do you have *cvs* on your system? You can quickly check this with:

```
[colin@desk]$ cvs --version
```

which should return some details, otherwise you will have to find your disc set and install it. The current version is 1.11.1p1.

You will need to know where the CVS repository is that you want to access – the project we're going to pick is the CVS project itself, which seems only fitting. This lives at *cvs.cvshome.org* in the */cvs* directory. It is to here that we need to *login*, usually using some sort of password server for cover. So:

```
[colin@desk]$ cvs -d
:pserver:guest@cvs.cvshome.org:/cvs login
```

will give us access, once we have entered the password, which will be prompted for, usually you just need to hit the Enter key. Almost always, anonymous access is granted for people to download the project's code, but the exact details of this can vary. Often the login will be *anonymous@* (instead of *guest@*, as in the example above) and the password will be blank, as above.

Depending on how often you are planning to take code from this CVS tree, you may want to consider setting the Environmental variable *$CVSROOT* for that tree, if you are only going to take code infrequently, then typing it in each time is not too great a burden. You need to save the parameter that is passed by the '–d' switch, in this case the *:pserver:guest@cvs.cvshome.org:/cvs*.

To allow you to download the project, which is called *ccsv* in this case, you use the *checkout* command

```
[colin@desk]$ cvs -d ➥
:pserver:guest@cvs.cvshome.org:/cvs ➥
checkout ccvs
```

If all is going well you will then get to see a listing of all the files that are now being downloaded, which get stored in a directory with the project name in the directory you ran the *cvs* command from, in this case */home/colin/ccvs*.

```
[colin@desk]$ cd ccvs
[colin@desk]$ ls
```

will give me a listing of all the files and subsequent directories in the 'cvs' project.

From here on, you are at the mercy of the project developers. Hopefully, if it is a good project, you will find that you now have some documentation that someone new to the project can read and understand. The important files to look out for are README and INSTALL. If there are not enough clues in these two files to take you further then you may be looking at a project that has yet to fully develop its documentation, either because it is small, or that no one wants to do the job. You still have the option of contacting the project developers by email, to ask for advice and, if you are let down here, then you at least know that the project isn't worth bothering with.

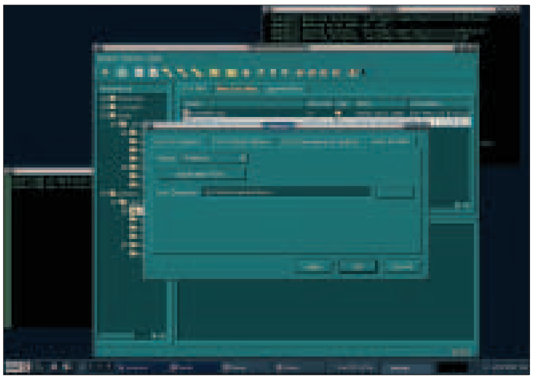
Sometimes, you may find that the project homepage has the option to allow you to 'browse' the cvs tree. If this is the case, it can sometimes be better to download some of the documentation files from this browsing session rather than download the whole tree only to find out that there is no documentation, or that it is in German, or some such.

Graphical front-ends

Graphical front-ends are available for CVS, to take away some of this command line misery.

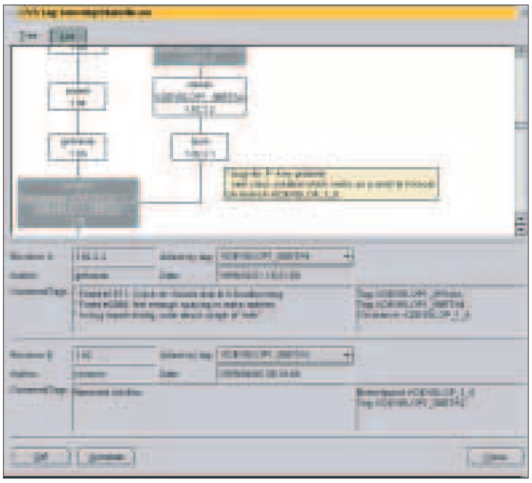
LinCVS

LinCVS is a German project, but thankfully with most pages also in English. It acts as a reliable graphical front-end for the CVS-client supporting both CVS versions 1.9 and 1.10, perhaps even older ones. It enables the user to check out, or download, a module from the CVS tree and import modules to the repository if you have been granted access rights. You can also update and retrieve the status of a working directory or single files and use common operations like add, remove and commit, diff against the repository or view of the log messages in list form, all of which is of much more use when you are actually contributing to the development of the project.



Accessing CVS trees with LinCVS

After selection of a start directory, the program enables you to automatically find all CVS projects contained in sub directories using the "Project-Explorer", and enables the user to add them and displays the corresponding tree on the desktop. All features can be accessed by pull-down menus or a simple click on the right mouse button and may be applied to both single and multiple files or complete trees.



Cervisia – with a clever tree display

Cervisia

Cervisia is another graphical front-end for CVS client-side actions, which has some really neat features like retrieving directories and single files and examining their status and a graphical *diff* utility which highlights the differences between different revisions in the same repository.

Getting involved

At best, this is supposed to be a two way process. Contributions are a start, but you may want to help further, with the documentation or, if you are up to it, the coding. CVS is used for uploading contributions and when you get to know a project's development team well enough, they might even give you your own account and password, but more of that will have to be left for another article.

Info

CVS homepage: *http://www.cvshome.org/*
The CVS homepage also contains a very good manual
LinCVS: *http://www.lincvs.org/*
Cervisia: *http://cervisia.sourceforge.net/*