CVS on Windows Platforms Made Easy   Oct 12, 2005   Charles Lohr

CVS stands for Concurrent Versions System. It is commonly used code management software
that is used in many open source projects. It allows many users to work on the same software at
the same time. It even allows multiple users to work on the same file at the same time.

Users can 'checkout' the current source for an entire project, modify their source to the way they
please, 'diff' their modifications to the current source and see what they modified on a line-by-
line basis if desired. If they wish for their change to be in the main copy of the source, they can
write up a brief explanation and 'commit' their change. At any time, users can update their
project to synchronize it with the current repository.

You can anonymously check out the CVS of any project on SourceForge.net. The CVSROOT
you need to use in order to check out their CVS can be found by visiting the project's page
SF.net, and clicking on CVS. You use an application called ViewCVS to look at an overview of
any project's source code. For example

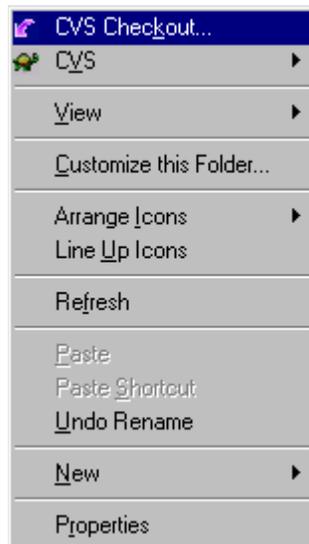   http://cvs.sourceforge.net/viewcvs.py/hgengine/

In order to use CVS, you need to install a CVS client. For Windows, there are two options. You
can either use Tortoise CVS or WinCVS. Tortoise CVS installs as extensions to explorer and is
somewhat easy to set up. WinCVS is a slightly more advanced program that does not install any
explorer extensions.

When you install your program, you will need to know your CVSROOT (from sf.net) and a
module name. If you are checking out a module anonymously, you will need to use :pserver:
authentication, otherwise (if you are logging in with your account) you will need to use :ext:
authentication. You will need to check out a given module from the project. You can find all the
modules that a project has by viewing its CVS. The folder names in the project root's folder
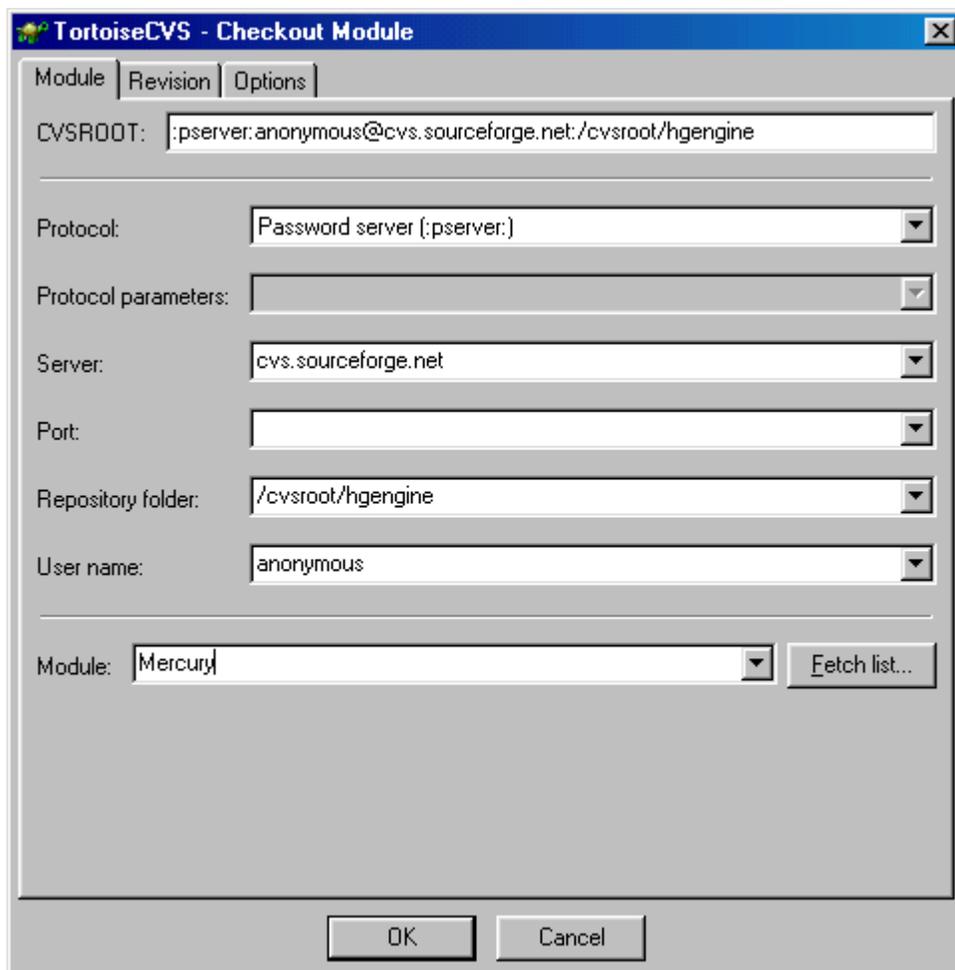contain

TortoiseCVS
You first need to download TortoiseCVS from http://sourceforge.net/projects/tortoisecvs the
package name is tortoisecvs-stable. When you install it, it will probably need to reboot. Next,
make a folder somewhere for the project you are checking out. Normally it's easiest to do this on
the root of your drive. Open that folder, right-click in the file area and select "CVS Checkout..."
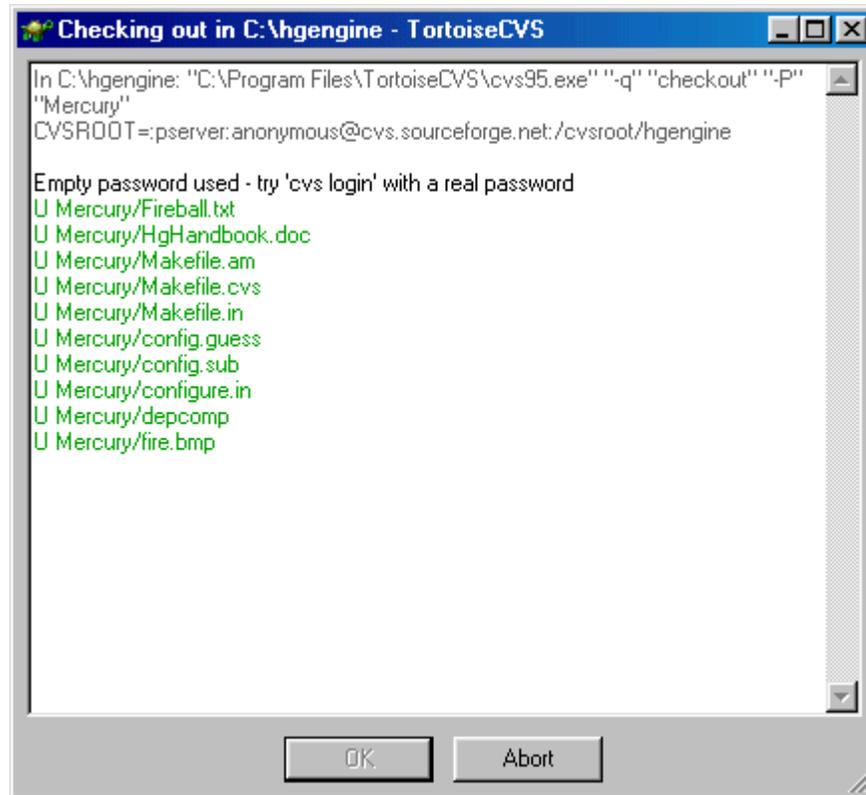like below:

A typical right-click popup menu with TortoiseCVS

Next, you will be presented with a prompt asking for you to input a CVSROOT (from SourceForge.net) and a Module Name. Both of these can be obtained by looking on SourceForge.net for the project you want to use.
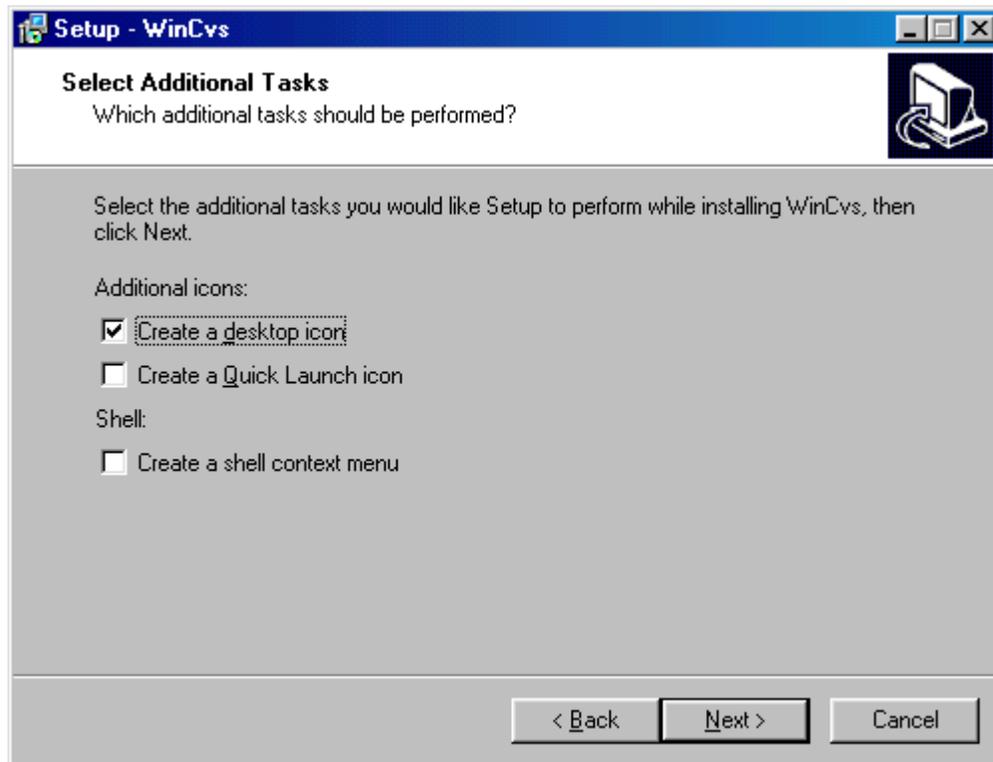

A typical TortoiseCVS checkout menu

When you click OK, Tortoise CVS will check out the repository to the folder you wanted to check out in. You can proceed to right-click on any of the files, or folders to perform any of the normal CVS functions.
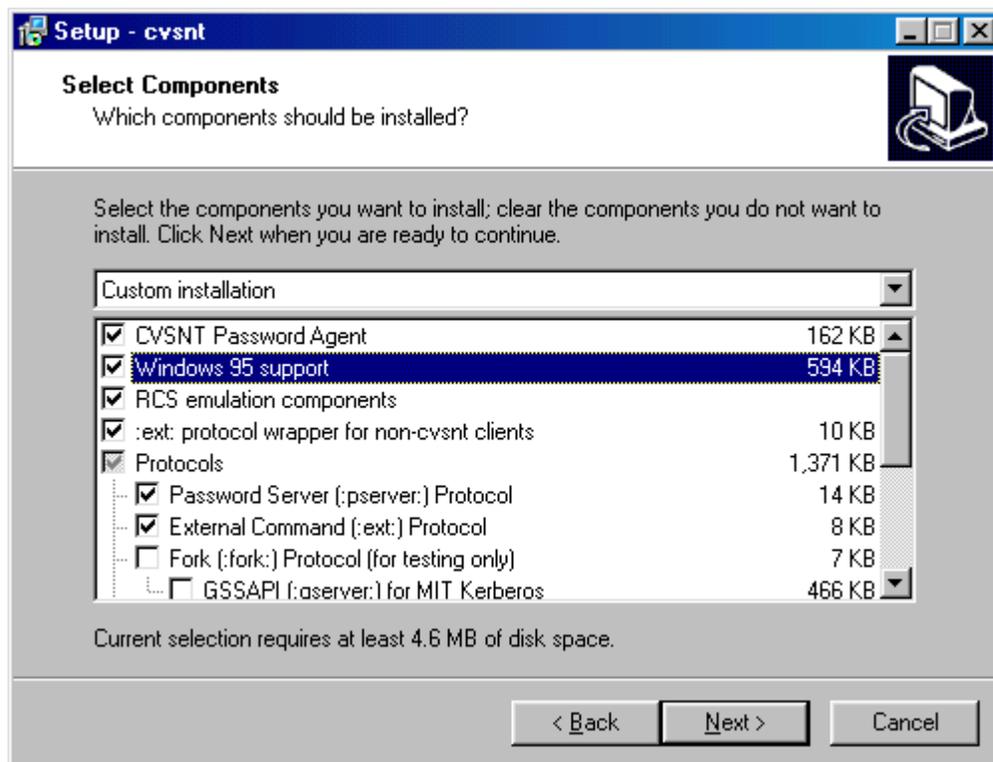

TortoiseCVS checking out

WinCVS

First, you need to download WinCVS from http://www.wincvs.org/. Open the .zip file, and install WinCVS and CVSnt with Windows 95 Support. You can chose to install the shell extensions to WinCVS but considering the reason most people use WinCVS is because they want to minimize CVS's interaction with their system it doesn't make all that much sense.
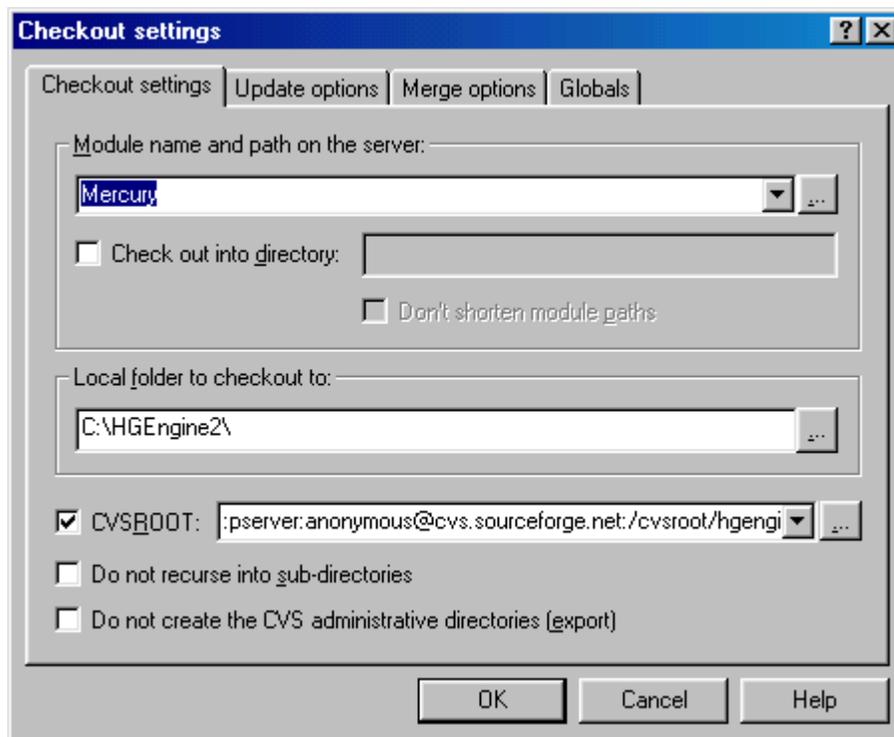
Unchecked Shell Context Menu

When you install CVSNT, you will need to install Windows95 support since that is what WinCVS and many other GUI-based CVS applications use to interface to CVSNT. CVSNT allows you also to use normal command-line (Linux-style) CVS.

Checked Windows 95 Support

Once both applications are installed, you will need to install an SSH client you can either use normal SSH, a modified version of Putty (distributed on this site as well as in TortoiseCVS) or Putty/Plink with a work-between program. No matter what method you choose, you will need to make sure that there is an SSH application by the name of ssh.exe somewhere within your computer's paths. If you want to install the modified putty, you can download ssh.exe from this website and save it to your Windows folder.

Once this is done, open WinCVS and go to the Remote menu and select Checkout Module just like before, put in your CVSROOT and Module name.

WinCVS checkout prompt

If you are going to log in to the server so you get commit access you will need to use the :ext: authentication method instead of :pserver:.