

What is CVS ?

The Concurrent Versioning System (CVS) is used to coordinate software development efforts among multiple developers. CVS provides a central storage for all source files, and handles changes via versioning.

How does CVS work ?

Prior to working, developers usually obtain a local copy of the most recent project status, their very own Sandbox Version. They are free to change their Sandbox files and check them into the version control system, which issues altered files a new version ID.

Other developers can access new versions of files once they have been submitted to the CVS. In order to update their local files, developers need to issue a CVS-Update.

In the case of several developers modifying the same file, the re-submission of this file to the version control system will fail for all but the first developer to check his file back into the CVS. Developers, whose check-in (i.e. terminology of CVS for re-submission of files) failed due to such versioning issues, have to resolve these.

This is generally done by checking-out (i.e. terminology of CVS for obtaining files) the most recent CVS version of a file and either let CVS merge it with the local file or in cases where this option is not available, apply manual modifications. For instance CVS cannot automatically merge files, where the same lines of code have been modified. Such conflicting lines are marked as follows:

```
<<<<<<< filename
  your changes
=====
  code merged from repository
>>>>>>> revision
```

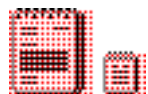
How does TortoiseCVS work ?

TortoiseCVS (<http://www.tortoisecvs.org>) is integrated in the Windows Explorer. All the files of the local Sandbox will be marked with overlays, which represent the status of the files :

- green square : the file matches the file of the CVS repository



- red square : the local file has been modified from the current CVS repository version



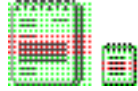
- purple square : the local file has a conflict with the version of the CVS repository



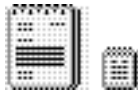
- question mark : the file is not in the CVS repository



- green square with red bar : the file is under CVS watch



- gray square : the file is ignored by the CVS



Checking in :

To check a completely new file in, select the file(s), right-click and select “CVS add”.

If you commit any files, you will be asked for a comment. Try to describe the changes precisely in this dialog.

Just commit important files (no binary...)

Checking out :

Select the files, right-click and select CVS Update.

Configuration of TortoiseCVS :

- Add the environment-variable “CVS_SERVER” with the value “/usr/local/dist/bin/cvs”
- select the protocol Secure Shell (:ext:)
- select the directory “/u/halle/sielhors/home_sun/.cvsroot/”
- select the module “DelivARy”

Working with TortoiseCVS without a password :

- download “puttygen.exe” and “pageant.exe” from <http://www.chiark.greenend.org.uk/~sgtatham/putty/>
- create a SSH2 RSA-Key with puttygen, protect it with a passphrase and save the private key. Add the public-key in the sunhalle in your home directory in the file /.ssh/authorized_keys (the string at the top of puttygen)
- start pageant and add the key to the list (you have to enter the passphrase)