

Technical details

Implementation / Requirements

CrossVC XXL is a c++ implementation using the qt library. It is based on the GPL'ed CrossVC open source project with extensions for commercial usage. It relies on the original and public available CVS client. CrossVC XXL is a CVS wrapper, and so does not provide its private CVS- implementation. It is the choice of the end user to update to another CVS version, maybe because a CVS- security-release requires this.

Required libraries and tools are:

Windows:

- all shipped with the CrossVC XXL package

Unix:

- cvs
- qt3 run time library, qt 3.6 recommended
- ssh (optionally, if cvs via ssh is desired)

Mac:

- all shipped with the CrossVC XXL package

CVS support:

CrossVC XXL /CVS:

- CVS 1.10 - 1.12.x CVS series
- Best tested with the current stable CVS- 1.11.x client.

CrossVC XXL /CVSNT:

- Smart merge support
- Extended log information
- Others - depending on the installed version

State tracking:

One of the outstanding features is the ability to always keep track with the current working copy state. Depending on the watch-level-state (see Performance / fine tuning), changes are recognized even if invoked from outside CrossVC XXL (from an ide, a CVS commandline client or whatever). The state is reflected graphical per file in the file view, and per directory in the workbench tree. The directory tree states are hierarchically organized, so a parent directory will reflect the state with the highest priority of all its child directories.

Available directory states are:

1. Directory is unchanged
2. Directory contains non-cvs-controlled files
3. Directory contains cvs-controlled, modified files
4. Directory contains uncommitted changes
5. Directory contains files that are not up-to-date
6. Directory contains files with conflicts

Performance / fine-tuning:

Watch-level:

CrossVC XXL can be fine-tuned to the users need. There are five watch-level states adjustable. These states are to detect changes in the working copy that are not triggered by CrossVC XXL. CrossVC XXL triggered events are always handled, independent of the watch level state.

Depending on the selected mode, CrossVC XXL will poll for states periodically. The period can be adjusted independent.

- Level 0: Big project mode, for projects with many (thousands) of files. User response time is reduced to a minimum for the price of more memory consumption and extensive internal caching. No polling is performed to detect non-cvs actions, a manual refresh is required in these cases.
- Level 1: Default mode, keeps track on the currently selected directory only. States are checked when switching to different directories.
- Level 2: Level 1 plus tracking of addition/removal of non-cvs-files in all directories of the currently selected project.
- Level 3: Level 2 plus checking all CVS- controlled files of current project for local modifications.
- Level 4: Level 3 plus checking all other projects in the workbench.

Project scanning:

If not started with a 'single directory startup option', CrossVC XXL will scan all projects in the workbench for setting up the current states. This can take an annoying amount of time on huge projects. In this case, CrossVC XXL can be set to on-the-fly-scanning mode. In this mode, only the first level directories are scanned on startup, deeper directories will only be scanned when first entered.

Projects, or parts of a project can be disabled in the workbench to keep them from being scanned, however recursive CVS commands will still affect these directories.

Special features:

CrossVC XXL aims towards professional users, and supports features not available with default CVS.

The most outstanding features are:

- Special-file-support (symlinks, independent from the CVS server/client version - currently unix only)
- Fine-grained merge/merge-preview dialog - shows all details of a merge and allows the user to decide what parts of the cvs-suggested merge should take place or be skipped.
- Repository browser, allows to browse the repository without checking out any files
- Repository tag browser, shows all kind of tags on the server side
- Import vendor branches. Required vendor tags are auto-detected, if more than one vendor-branch is used, a selection is supplied. Multiple vendor branches are supported.
- Conditional edits, as an alternative to reserved checkouts. Checks for other editors and only edits if no-one else edits the file.
- Non-modal dialogs. Most dialogs are non-modal, so multiple differences, project trees, file trees or alike can be looked at while typing a commit

- message.
- External diff/merge tools are supported. An unlimited number of tools can be specified per pattern or regular expression.
 - Sub-project support. CrossVC allows to drop several CVS- projects, dependent or independent, into one parent directory. If this parent directory is added to the workbench, all toplevel directories of independent CVS- projects will be marked with an anchor, dependent projects will be integrated automatically.
 - Skip non-controlled directories. If a project is added to the workbench, its whole tree will be scanned for further CVS projects, even if there are intermediate non-cvs directories. This can be switched off if required.

NCVS extension:

The NCVS binary is an extension to CVS. If additionally installed on the CVS server, clients can be connected using the same connection mechanism CVS uses. Upon connection, NCVS informs the client about edit/unedit/commit/release actions of other users.

In combination with CrossVC XXL, files are continuously monitored for actions of other users. The CrossVC XXL file view will show the latest current editors login name in the 'Last editor' column. In case one of the above mentioned actions takes place, the user is informed by a balloon message or popup dialog, depending on customization.

NVCS will not work offline as it requires a persistent connection.