

CVS Quick Reference Card

Andrew Ford refcards.com™
CVS is the Concurrent Versions System, the dominant open-source,
network-transparent version control system.

Command Line Syntax

cvcs [global_opts] command [command_opts] [command_args]

Global options:

--allow-root=rootdir
Specify legal CVSROOT directory. (server only)
-a Authenticate all communication. (client only)
-d root Specify the CVSROOT.
-e editor Edit messages with editor.
-f Do not read the ~/.cvsrc file.
-H Print a help message.
--help ditto
-l Do not log in CVSROOT/history file.
-n Do not change any files.
-Q Be really quiet.
-q Be somewhat quiet.
-r Make new working files read-only.
-s var=value Set a user variable.
-T tempdir Put temporary files in tempdir.
-t Trace CVS execution.
-v Display CVS version and copyright information.
--version ditto
-w Make new working files read-write.
-x Encrypt all communication. (client only)
-z gzip-lvl Set the compression level. (client only)

CVS Commands

add Add new files or directories.
admin Repository administration.
annotate Display revision for each line.
checkout Check out sources for editing.
commit Commit changes into repository.
diff Show differences between revisions.
edit Prepare to edit a file being watched.
editors List users working on a file.
export Export files from CVS.
history Show repository access history.
import Import files into CVS.
init Create a CVS repository if none exists.
log Print log information for files.
login Prompt for password for client access.
logout Removed stored password.
rdiff Show differences between releases.
release Indicate that a directory is no longer in use.
remove Remove an entry from the repository.
rtag Add a symbolic tag to a module.
status Display status information.
tag Add a symbolic tag to checked-out files.
unedit Undo an edit command.
update Bring work tree into sync with repository.
watch Tell CVS to provide notifications.
watchers List who is watching a file.

Keywords

\$Author\$ Login name of the user who checked in the revision
\$Date\$ Date and time (UTC) the revision was checked in
\$Header\$ Standard header with full pathname of file in repository
\$Id\$ Standard header with filename of file in repository
\$Locker\$ Login name of the user who locked the revision
\$Log\$ Log message supplied on commit
\$Name\$ Tag name used to check out the file
\$RCSfile\$ Name of the file in repository (without path)
\$Revision\$ Revision number
\$Source\$ Full pathname of file in repository
\$State\$ State assigned to revision
Keyword expansion modes:
-kkv Generate keyword strings in default format
-kkv1 Like -kkv except locker's name is inserted
-kk Generate only keyword names in keyword strings
-kv Generate old keyword string present in the working file just before it was checked in.
-ko no expansion
-kb no expansion, file is binary

Commands, command options and arguments

add [options] [files]...

Adds a new file/directory
-k kflag Set keyword expansion
-m msg Set file description

admin [options] [files]...

Administration of history files in the repository.
-b [rev] Set default branch
-c string Set comment leader
-k subst Set keyword substitution
-l [rev] Lock revision rev, or latest revision
-m rev:msg replace the log message of revision rev with msg
-o range Delete revisions from the repository.
-q Run quietly; do not print diagnostics.
-s Set the state.
state[:rev]
-t Set file description from standard input
-t file Set file description from file
-t -string Set file description to string
-u [rev] Unlock revision rev, or latest revision

annotate [options] files...

Show last revision where each line was modified.
-D date Annotate the most recent revision no later than date
-f Use head revision if tag/date not found
-l Local – run only in current working directory
-R Operate recursively (default)
-r tag Annotate revision tag

checkout [options] modules...

Get a copy of the sources.

-A Reset any sticky tags/date/options
-c Output the module database
-D date Check out revisions as of date (sticky)
-d dir Check out into dir
-f Use head revision if tag/date not found
-j rev Merge in changes
-k kflag Use kflag keyword expansion
-l Local – run only in current working directory
-N Do not 'shorten' module paths if -d specified
-n Do not run module program (if any)
-P Prune empty directories
-p Check out files to standard output (avoids stickiness)
-R Operate recursively (default)
-r tag Checkout revision tag (is sticky)
-s Like -c, but include module status

commit [options] [files...]

Check changes into the repository.
-F file Read log message from file
-f Force the file to be committed; disables recursion
-l Local – run only in current working directory
-m msg Use msg as log message
-n Do not run module program (if any)
-R Operate recursively (default)
-r rev Commit to rev

diff [options] files...

Show differences between revisions. In addition to the options shown below, accepts a wide variety of options to control output style, for example -c for context diffs.
-D date1 Diff revision for date against working file.
-D date2 Diff rev1/date1 against date2
-l Local – run only in current working directory
-N Include diffs for added and removed files
-R Operate recursively (default)
-r rev1 Diff revision for rev1 against working file
-r rev2 Diff rev1/date1 against rev2

edit [options] [files...]

Get ready to edit a watched file.
-a actions Specify actions for temporary watch, where actions is edit, unedit, commit, all, or none
-l Local – run only in current working directory
-R Operate recursively (default)

editors [options] [files...]

See who is editing a watched file.
-l Local – run only in current working directory
-R Operate recursively (default)

export [options] modules...

Export files from CVS.

-D *date* Check out revisions as of *date*
 -d *dir* Check out into *dir*
 -f Use head revision if tag/date not found.
 -k *kflag* Use *kflag* keyword expansion
 -l Local – run only in current working directory
 -N Do not “shorten” module paths if -d specified
 -n Do not run module program (if any)
 -P Prune empty directories
 -R Operate recursively (default)
 -r *tag* Checkout revision *tag*

history [options] [files...]

Show repository access history.

-a All users (default is self)
 -b *str* Back to record with *str* in module/file/repos field
 -c Report on committed (modified) files
 -D *date* Since *date*
 -e Report on all record types
 -l Last modified (committed or modified report)
 -m *module* Report on module (repeatable)
 -n *module* In module
 -o Report on checked out modules
 -r *rev* Since revision *rev*
 -T Produce report on all tags
 -t *tag* Since *tag* record placed in history file (by anyone)
 -u *user* For user *user* (repeatable)
 -w Working directory must match
 -x *types* Report on *types*, one or more of TOEFWUCGMAR
 -z *zone* Output for time zone *zone*

import [options] repository vendor-tag release-tags...

Import files into CVS, using vendor branches.

-b *branch* Import to vendor branch *branch*
 -d Use the file’s modification time as the time of import.
 -k *kflag* Set default keyword substitution mode
 -m *msg* Use *msg* for log message
 -I *ign* More files to ignore (! to reset)
 -W *spec* More wrappers

init

Create a CVS repository if it doesn’t exist.

log [options] [files...]

Print out history information for files.

-b Only list revisions on the default branch.
 -d *dates* Specify dates (*d1* ; *d2* for range, *d* for latest before)
 -h Only print header
 -l Local – run only in current working directory
 -N Do not list tags
 -R Only print name of RCS file
 -r *revs* Only list revisions *revs*
 -s *states* Only list revisions with specified states.
 -t Only print header and descriptive text
 -w *logins* Only list revisions checked in by specified logins

login

Prompt for password for authenticating server.

logout

Remove stored password for authenticating server.

rdiff [options] modules...

Show differences between releases.

-c Context diff output format (default)
 -D *date* Select revisions based on *date*
 -f Use head revision if tag/date not found
 -l Local – run only in current working directory
 -R Operate recursively (default)
 -r *rev* Select revisions based on *rev*
 -s Short patch – one liner per file
 -t Top two diffs – last change made to the file
 -u Unidiff output format

release [options] directory

Indicate that a directory is no longer in use.

-d Delete the given directory

remove [options] [files...]

Remove an entry from the repository.

-f Delete the file before removing it
 -l Local – run only in current working directory
 -R Operate recursively (default)

rtag [options] tag modules...

Add a symbolic tag to a module.

-a Clear tag from removed files that would not otherwise be tagged
 -b Create a branch named tag
 -D *date* Tag revisions as of *date*
 -d Delete tag
 -F Move tag if it already exists
 -f Force a head revision match if tag/date not found
 -l Local – run only in current working directory
 -n No execution of tag program
 -R Operate recursively (default)
 -r *rev* Tag existing tag *rev*

status [options] files...

Display status information in a working directory.

-l Local – run only in current working directory.
 -R Operate recursively (default).
 -v Include tag information for file.

tag [options] tag [files...]

Add a symbolic tag to checked out version of files.

-b Create a branch named tag.
 -c Check that working files are unmodified.
 -D *date* Tag revisions as of *date*.
 -d Delete tag.
 -F Move tag if it already exists.
 -f Force a head revision match if tag/date not found.
 -l Local – run only in current working directory.
 -R Operate recursively (default).
 -r *rev* Tag existing tag *rev*.

unedit [options] [files...]

Undo an edit command.

-a *actions* Specify actions for temporary watch, where *actions* is edit, unedit, commit, all, or none.
 -l Local – run only in current working directory.
 -R Operate recursively (default).

update [options] [files...]

Bring work tree in sync with repository.

-A Reset any sticky tags/date/options.
 -D *date* Check out revisions as of *date* (is sticky).
 -d Create directories.
 -f Use head revision if tag/date not found.
 -I *ign* More files to ignore (! to reset).
 -j *rev* Merge in changes.
 -k *kflag* Use *kflag* keyword expansion.
 -l Local – run only in current working directory.
 -P Prune empty directories.
 -p Check out files to standard output (avoids stickiness).
 -R Operate recursively (default).
 -r *tag* Checkout revision *tag* (is sticky).
 -W *spec* More wrappers.

watch {on|off|add|remove} [options] [files...]

Set notifications for files.

{on|off} Turn on/off read-only checkouts of files.
 {add|remove} Add or remove notification on actions.
 -a *actions* Specify actions for temporary watch, where *actions* is edit, unedit, commit, all, or none.
 -l Local – run only in current working directory.
 -R Operate recursively (default).

watchers [options] [files...]

See who is watching a file.

-l Local – run only in current working directory.
 -R Operate recursively (default).

Resources

<http://www.cvshome.com>
<http://www.refcards.com>

CVS home page
 Quick reference cards

CVS Quick Reference Card

Revision 0.1 (draft) for CVS version 1.10

[October 2000]

A [refcards.com](http://www.refcards.com)™ quick reference card

The content of this card is in the public domain.

Design © 2000 Ford & Mason Ltd. All rights reserved.

Permission is granted to print and duplicate this card for personal or individual, internal business use. Copies of this card (& others) can be ordered through our web site: <http://www.refcards.com>, which also has versions available for downloading.

Please send feedback to: feedback@refcards.com

[refcards.com](http://www.refcards.com) is a trademark of Ford & Mason Ltd.