

**NAME**

rsclean - clean up working files

**SYNOPSIS**

**rsclean** [*options*] [*file ...*]

**DESCRIPTION**

**rsclean** removes files that are not being worked on. **rsclean -u** also unlocks and removes files that are being worked on but have not changed.

For each *file* given, **rsclean** compares the working file and a revision in the corresponding RCS file. If it finds a difference, it does nothing. Otherwise, it first unlocks the revision if the **-u** option is given, and then removes the working file unless the working file is writable and the revision is locked. It logs its actions by outputting the corresponding **rcs -u** and **rm -f** commands on the standard output.

Files are paired as explained in **ci(1)**. If no *file* is given, all working files in the current directory are cleaned. Filenames matching an RCS suffix denote RCS files; all others denote working files.

The number of the revision to which the working file is compared may be attached to any of the options **-n**, **-q**, **-r**, or **-u**. If no revision number is specified, then if the **-u** option is given and the caller has one revision locked, **rsclean** uses that revision; otherwise **rsclean** uses the latest revision on the default branch, normally the root.

**rsclean** is useful for **clean** targets in makefiles. See also **rscdiff(1)**, which prints out the differences, and **ci(1)**, which normally reverts to the previous revision if a file was not changed.

**OPTIONS****-ksubst**

Use *subst* style keyword substitution when retrieving the revision for comparison. See **co(1)** for details.

**-n[rev]**

Do not actually remove any files or unlock any revisions. Using this option will tell you what **rsclean** would do without actually doing it.

**-q[rev]**

Do not log the actions taken on standard output.

**-r[rev]**

This option has no effect other than specifying the revision for comparison.

- T** Preserve the modification time on the RCS file even if the RCS file changes because a lock is removed. This option can suppress extensive recompilation caused by a **make(1)** dependency of some other copy of the working file on the RCS file. Use this option with care; it can suppress recompilation even when it is needed, i.e. when the lock removal would mean a change to keyword strings in the other working file.
  
- u[*rev*]**  
Unlock the revision if it is locked and no difference is found.
  
- V** Print RCS's version number.
  
- V*n***  
Emulate RCS version *n*. See **co(1)** for details.
  
- x*suffixes***  
Use *suffixes* to characterize RCS files. See **ci(1)** for details.
  
- z*zone***  
Use *zone* as the time zone for keyword substitution; see **co(1)** for details.

## EXAMPLES

```
rcsclean *.c *.h
```

removes all working files ending in **.c** or **.h** that were not changed since their checkout.

```
rcsclean
```

removes all working files in the current directory that were not changed since their checkout.

## FILES

**rcsclean** accesses files much as **ci(1)** does.

## ENVIRONMENT

### RCSINIT

Options prepended to the argument list, separated by spaces. A backslash escapes spaces within an option. The **RCSINIT** options are prepended to the argument lists of most RCS commands.

Useful **RCSINIT** options include **-q**, **-V**, **-x**, and **-z**.

### RCS\_MEM\_LIMIT

Normally, for speed, commands either memory map or copy into memory the RCS file if its size is

less than the *memory-limit*, currently defaulting to “unlimited”. Otherwise (or if the initially-tried speedy ways fail), the commands fall back to using standard i/o routines. You can adjust the memory limit by setting **RCS\_MEM\_LIMIT** to a numeric value *lim* (measured in kilobytes). An empty value is silently ignored. As a side effect, specifying **RCS\_MEM\_LIMIT** inhibits fall-back to slower routines.

## **TMPDIR**

Name of the temporary directory. If not set, the environment variables **TMP** and **TEMP** are inspected instead and the first value found is taken; if none of them are set, a host-dependent default is used, typically **/tmp**.

## **DIAGNOSTICS**

The exit status is zero if and only if all operations were successful. Missing working files and RCS files are silently ignored.

## **IDENTIFICATION**

Author: Walter F. Tichy.

Manual Page Revision: 5.10.1; Release Date: 2022-02-03.

Copyright (C) 2010-2022 Thien-Thi Nguyen.

Copyright (C) 1990, 1991, 1992, 1993 Paul Eggert.

Copyright (C) 1982, 1988, 1989 Walter F. Tichy.

## **SEE ALSO**

**ci(1)**, **co(1)**, **ident(1)**, **rsc(1)**, **rscdiff(1)**, **rscmerge(1)**, **rlog(1)**, **rscfile(5)**.

Walter F. Tichy, RCS--A System for Version Control, *Software--Practice & Experience* **15**, 7 (July 1985), 637-654.

The full documentation for RCS is maintained as a Texinfo manual. If the **info(1)** and RCS programs are properly installed at your site, the command

**info rcs**

should give you access to the complete manual. Additionally, the RCS homepage:

**<http://www.gnu.org/software/rcs/>**

has news and links to the latest release, development site, etc.

## **BUGS**

At least one *file* must be given in older Unix versions that do not provide the needed directory scanning operations.