NAME

rcs − change RCS file attributes

SYNOPSIS

rcs *options file* . . .

DESCRIPTION

rcs creates new RCS files or changes attributes of existing ones. An RCS file contains multiple revisions of text, an access list, a change log, descriptive text, and some control attributes. For **rcs** to work, the caller's login name must be on the access list, except if the access list is empty, the caller is the owner of the file or the superuser, or the **−i** option is present.

Pathnames matching an RCS suffix denote RCS files; all others denote working files. Names are paired as explained in **ci**(1). Revision numbers use the syntax described in **ci**(1).

OPTIONS

−i Create and initialize a new RCS file, but do not deposit any revision. If the RCS file has no path prefix, try to place it first into the subdirectory **./RCS**, and then into the current directory. If the RCS file already exists, print an error message.

−a*logins*
: Append the login names appearing in the comma-separated list *logins* to the access list of the RCS file.

−A*oldfile*
: Append the access list of *oldfile* to the access list of the RCS file.

−e[*logins*]
: Erase the login names appearing in the comma-separated list *logins* from the access list of the RCS file. If *logins* is omitted, erase the entire access list.

−b[*rev*] Set the default branch to *rev*. If *rev* is omitted, the default branch is reset to the (dynamically) highest branch on the trunk.

−c*string*
: Set the comment leader to *string*. An initial **ci**, or an **rcs −i** without **−c**, guesses the comment leader from the suffix of the working filename.

This option is obsolescent, since RCS normally uses the preceding **Log** line's prefix when inserting log lines during checkout (see **co**(1)). However, older versions of RCS use the comment leader instead of the **Log** line's prefix, so if you plan to access a file with both old and new versions of RCS, make sure its comment leader matches its **Log** line prefix.

−k*subst*
: Set the default keyword substitution to *subst*. The effect of keyword substitution is described in **co**(1). Giving an explicit **−k** option to **co**, **rcsdiff**, and **rcsmerge** overrides this default. Beware **rcs −kv**, because **−kv** is incompatible with **co −l**. Use **rcs −kkv** to restore the normal default keyword substitution.

−l[*rev*] Lock the revision with number *rev*. If a branch is given, lock the latest revision on that branch. If *rev* is omitted, lock the latest revision on the default branch. Locking prevents overlapping changes. If someone else already holds the lock, the lock is broken as with **rcs −u** (see below).

−u[*rev*] Unlock the revision with number *rev*. If a branch is given, unlock the latest revision on that branch. If *rev* is omitted, remove the latest lock held by the caller. Normally, only the locker of a revision can unlock it. Somebody else unlocking a revision breaks the lock. This causes a mail message to be sent to the original locker. The message contains a commentary solicited from the breaker. The commentary is terminated by end-of-file or by a line containing **.** by itself.

−L Set locking to *strict*. Strict locking means that the owner of an RCS file is not exempt from locking for checkin. This option should be used for files that are shared.

–U Set locking to non-strict. Non-strict locking means that the owner of a file need not lock a revision for checkin. This option should *not* be used for files that are shared. Whether default locking is strict is determined by your system administrator, but it is normally strict.

–m*rev***:***msg*
Replace revision *rev*'s log message with *msg*.

–M Do not send mail when breaking somebody else's lock. This option is not meant for casual use; it is meant for programs that warn users by other means, and invoke **rcs –u** only as a low-level lock-breaking operation.

–n*name*[**:**[*rev*]]
Associate the symbolic name *name* with the branch or revision *rev*. Delete the symbolic name if both **:** and *rev* are omitted; otherwise, print an error message if *name* is already associated with another number. If *rev* is symbolic, it is expanded before association. A *rev* consisting of a branch number followed by a **.** stands for the current latest revision in the branch. A **:** with an empty *rev* stands for the current latest revision on the default branch, normally the trunk. For example, **rcs –n***name***: RCS/*** associates *name* with the current latest revision of all the named RCS files; this contrasts with **rcs –n***name***:$ RCS/*** which associates *name* with the revision numbers extracted from keyword strings in the corresponding working files.

–N*name*[**:**[*rev*]]
Act like **–n**, except override any previous assignment of *name*.

–o*range*
deletes ("outdates") the revisions given by *range*. A range consisting of a single revision number means that revision. A range consisting of a branch number means the latest revision on that branch. A range of the form *rev1***:***rev2* means revisions *rev1* to *rev2* on the same branch, **:***rev* means from the beginning of the branch containing *rev* up to and including *rev*, and *rev***:** means from revision *rev* to the end of the branch containing *rev*. None of the outdated revisions can have branches or locks.

–q Run quietly; do not print diagnostics.

–I Run interactively, even if the standard input is not a terminal.

–s*state*[**:***rev*]
Set the state attribute of the revision *rev* to *state*. If *rev* is a branch number, assume the latest revision on that branch. If *rev* is omitted, assume the latest revision on the default branch. Any identifier is acceptable for *state*. A useful set of states is **Exp** (for experimental), **Stab** (for stable), and **Rel** (for released). By default, **ci**(1) sets the state of a revision to **Exp**.

–t[*file*] Write descriptive text from the contents of the named *file* into the RCS file, deleting the existing text. The *file* pathname cannot begin with **–**. If *file* is omitted, obtain the text from standard input, terminated by end-of-file or by a line containing **.** by itself. Prompt for the text if interaction is possible; see **–I**. With **–i**, descriptive text is obtained even if **–t** is not given.

–t–*string*
Write descriptive text from the *string* into the RCS file, deleting the existing text.

–T Preserve the modification time on the RCS file unless a revision is removed. This option can suppress extensive recompilation caused by a **make**(1) dependency of some 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 a change to the RCS file would mean a change to keyword strings in the working file.

–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 default time zone. This option has no effect; it is present for compatibility with other RCS commands.

At least one explicit option must be given, to ensure compatibility with future planned extensions to the **rcs** command.

COMPATIBILITY

The **−b***rev* option generates an RCS file that cannot be parsed by RCS version 3 or earlier.

The **−k***subst* options (except **−kkv**) generate an RCS file that cannot be parsed by RCS version 4 or earlier.

Use **rcs −V***n* to make an RCS file acceptable to RCS version *n* by discarding information that would confuse version *n*.

RCS version 5.5 and earlier does not support the **−x** option, and requires a **,v** suffix on an RCS pathname.

FILES

rcs accesses files much as **ci**(1) does, except that it uses the effective user for all accesses, it does not write the working file or its directory, and it does not even read the working file unless a revision number of **$** is specified.

ENVIRONMENT

RCSINIT

options prepended to the argument list, separated by spaces. See **ci**(1) for details.

DIAGNOSTICS

The RCS pathname and the revisions outdated are written to the diagnostic output. The exit status is zero if and only if all operations were successful.

IDENTIFICATION

Author: Walter F. Tichy.
Manual Page Revision: 5.13; Release Date: 1995/06/05.

SEE ALSO

rcsintro(1), co(1), ci(1), ident(1), rcsclean(1), rcsdiff(1), rcsmerge(1), rlog(1), rcsfile(5)
Walter F. Tichy, RCS—A System for Version Control, *Software—Practice & Experience* **15**, 7 (July 1985), 637-654.

BUGS

A catastrophe (e.g. a system crash) can cause RCS to leave behind a semaphore file that causes later invocations of RCS to claim that the RCS file is in use. To fix this, remove the semaphore file. A semaphore file's name typically begins with **,** or ends with **_**.

The separator for revision ranges in the **−o** option used to be **−** instead of **:**, but this leads to confusion when symbolic names contain **−**. For backwards compatibility **rcs −o** still supports the old **−** separator, but it warns about this obsolete use.

Symbolic names need not refer to existing revisions or branches. For example, the **−o** option does not remove symbolic names for the outdated revisions; you must use **−n** to remove the names.