Appendix A. The Five−Minute RCS Tutorial

Contents:

References for More Information

This quick tutorial will teach you everything you need to know about how to use Revision Control System (RCS) for system administration. RCS has considerably more functionality than we'll discuss here, so be sure to take a look at the manual pages and the reference at the end of this appendix if you plan to use it heavily.

RCS functions like a car rental agency. Only one person at a time can actually rent a particular car and drive it off the lot. A new car can only be rented after the agency has added it to their pool. Customers can browse the list of cars (and their features) at any time, but if two people want to rent the same car, the second must wait for the car to be returned to the lot before renting it. Finally, car rental agencies inspect cars very carefully after they have been returned and record any changes to the car during the rental. All of these properties hold true for RCS as well.

In RCS, a file is like a car. If you wish to keep track of a file using RCS (i.e., add it to the rental lot) you "check it in" for the first time:

```
$ ci -u filename
```

ci stands for "check in," and the *−u* tells RCS to leave the file in place during the check−in. When a file is checked in (i.e., made available for rental), RCS does one of two things to remind the user that the file is under RCS's control:

1. Deletes the original file, leaving only the RCS archive file behind. This archive file is usually called *filename,v* and is either kept in the same directory as the original file or in a subdirectory called *RCS* (if the user creates it).

2. If *−u* is used as we showed above, it checks the file out again, leaving the permissions on the file to be "read−only."

To modify a file under RCS's control (i.e., rent a car), you first need to "check−out" that file:

```
$ co -l filename
```

The **`-l`** switch tells RCS to "strictly lock" that file (i.e., do not allow any other user to check out the file at the same time). Other switches that are commonly used with *co* are:

- *−r <revision number>:* to check out an older revision of a file.

- *−p:* to print a past revision to the screen without actually checking it out.

Once you are done modifying a file, you need to check it back in using the same command we used above to put the file under RCS in the first place (*ci −u filename*). The check−in process stores any changes made to this file in a space−efficient manner.

Each time a file that has been modified is checked in, it is given a new revision number. At check−in time, RCS will prompt you for a comment to be placed in the change log it automatically keeps for each file. This log and the listing of the current person who has checked out a file can be viewed using *rlog filename*.

If someone neglects to check their changes to a particular file back into RCS (e.g., they've gone home for the day and you have a real need to change the file yourself), you can break their lock using *rcs−u filename*. This command will prompt for a break−lock message that is mailed to the person who owns the lock.

After breaking the lock, you should check to see how the current copy differs from the RCS archive revision. *rcsdiff filename* will show you this information. If you wish to preserve these changes, check the file in (with an appropriate change−log comment), and then check it back out again before working on it. `rcsdiff`, like our `co` example above, can also take a *−r <revision number>* flag to allow you to compare two past revisions.

Table A−1 lists some command RCS operations and their command lines.

Table A−1. Common RCS Operations

RCS Operation	Command Line
Initial check−in of file (leaving file active in filesystem)	*ci −u filename*
Check out with lock	*co −l filename*
Check in and unlock (leaving file active in filesystem)	*ci −u filename*
Display version *x.y* of a file	*co −px.y filename*
Undo to version *x.y* (overwrites file active in filesystem with the specified revision)	*co −rx.y filename*
Diff file active in filesystem and last revision	*rcsdiff filename*
Diff versions *x.y* and *x.z*	*rcsdiff −rx.y −rx.z filename*
View log of checkins	*rlog filename*
Break an RCS lock held by another person on a file	*rcs −u filename*

Believe it or not, this is really all you need to get started using RCS. Once you start using it for system administration, you'll find it pays off handsomely.

A.1. References for More Information

ftp://ftp.gnu.org/pub/gnu/rcs has the latest source code for the RCS package.

Applying RCS and SCCS: From Source Control to Project Control, by Don Bolinger and Tan Bronson (O'Reilly, 1995).

http://www.sourcegear.com/CVS is where to go if you find you need features not found in RCS. The next step up is the very popular Concurrent Versions System (CVS). This is its main distribution point.

PREVIOUS	HOME	NEXT
10.7. References for More Information	BOOK INDEX	B. The Ten−Minute LDAP Tutorial