NAME

rcorder - print a dependency ordering of interdependent files

SYNOPSIS

rcorder [**-gp**] [**-k** *keep*] [**-s** *skip*] *file* ...

DESCRIPTION

The **rcorder** utility is designed to print out a dependency ordering of a set of interdependent files. Typically it is used to find an execution sequence for a set of shell scripts in which certain files must be executed before others.

Each file passed to **rcorder** must be annotated with special lines (which look like comments to the shell) which indicate the dependencies the files have upon certain points in the sequence, known as "conditions", and which indicate, for each file, which "conditions" may be expected to be filled by that file.

Within each file, a block containing a series of 'REQUIRE', 'PROVIDE', 'BEFORE' and 'KEYWORD' lines must appear. The format of the lines is rigid. Each line must begin with a single '#', followed by a single space, followed by 'PROVIDE:', 'REQUIRE:', 'BEFORE:', or 'KEYWORD:'. No deviation is permitted. Each dependency line is then followed by a series of conditions, separated by whitespace. Multiple 'PROVIDE', 'REQUIRE', 'BEFORE' and 'KEYWORD' lines may appear, but all such lines must appear in a sequence without any intervening lines, as once a line that does not follow the format is reached, parsing stops.

The options are as follows:

-g Produce a GraphViz (.dot) of the complete dependency graph instead of plaintext calling order list.

-k *keep* Add the specified keyword to the "keep list". If any **-k** option is given, only those files containing the matching keyword are listed. This option can be specified multiple times.

-p Generate ordering suitable for parallel startup, placing files that can be executed simultaneously on the same line.

-s *skip* Add the specified keyword to the "skip list". If any **-s** option is given, files containing the matching keyword are not listed. This option can be specified multiple times.

An example block follows:

```
# REQUIRE: networking syslog
# REQUIRE: usr
# PROVIDE: dns nscd
```

This block states that the file in which it appears depends upon the 'networking', 'syslog', and 'usr' conditions, and provides the 'dns' and 'nscd' conditions.

A file may contain zero 'PROVIDE' lines, in which case it provides no conditions, and may contain zero 'REQUIRE' lines, in which case it has no dependencies. There must be at least one file with no dependencies in the set of arguments passed to **rcorder** in order for it to find a starting place in the dependency ordering.

KEYWORDS

There are several *KEYWORDs* in use:

firstboot, **nojail**, **nojailvnet**, **nostart**
Used by rc(8).

suspend, **resume**
Used by **/etc/rc.suspend** and **/etc/rc.resume** (see acpiconf(8))

shutdown Used by rc.shutdown(8).

EXAMPLES

Print the dependency ordering of the services from the base system and ports(7):

```
$ rcorder /etc/rc.d/* /usr/local/etc/rc.d/*
```

Count the number of services in the base system, which specify the **nostart** keyword, while skipping those with **firstboot** and **nojailvnet**:

```
$ rcorder -k nostart -s firstboot -s nojailvnet /etc/rc.d/*  | wc -l
    3
```

DIAGNOSTICS

The **rcorder** utility may print one of the following error messages and exit with a non-zero status if it encounters an error while processing the file list.

Requirement %s in file %s has no providers. No file has a 'PROVIDE' line corresponding to a condition present in a 'REQUIRE' line in another file.

Circular dependency on provision %s in file %s. A set of files has a circular dependency which was detected while processing the stated condition. Loop visualization follows this message.

Circular dependency on file %s. A set of files has a circular dependency which was detected while processing the stated file.

%s was seen in circular dependencies for %d times. Each node that was a part of circular dependency loops reports total number of such encounters. Start with files having biggest counter when fighting with broken dependencies.

DIAGNOSTICS WITH GRAPHVIZ

Direct dependency is drawn with solid line, 'BEFORE' dependency is drawn as a dashed line. Each node of a graph represents an item from 'PROVIDE' lines. In case there are more than one file providing an item, a list of filenames shortened with basename(3) is shown. Shortened filenames are also shown in case 'PROVIDE' item does not match file name.

Edges and nodes where circular dependencies were detected are drawn bold red. If a file has an item in 'REQUIRE' or in 'BEFORE' that could not be provided, this missing provider and the requirement will be drawn bold red as well.

SEE ALSO

acpiconf(8), rc(8), rc.shutdown(8), service(8)

HISTORY

The **rcorder** utility appeared in NetBSD 1.5. **rcorder** utility first appeared in FreeBSD 5.0.

AUTHORS

Written by Perry E. Metzger <*perry@piermont.com*> and Matthew R. Green <*mrg@eterna.com.au*>.

BUGS

The 'REQUIRE' keyword is misleading: It does not describe which daemons have to be running before a script will be started. It describes which scripts must be placed before it in the dependency ordering. For example, if your script has a 'REQUIRE' on 'sshd', it means the script must be placed after the 'sshd' script in the dependency ordering, not necessarily that it requires **sshd** to be started or enabled.