

---

# Chapter 4

## Utility Programs

---

### Chapter Overview

#### finger

- Name
- Command Line
- Description
- Options

#### ftp

- Name
- Command Line
- Description
- Options
- Commands
- Aborting a File Transfer
- File Naming Conventions
- File Transfer Parameters

#### lpq

- Name
- Command Line
- Description
- Options
- Files
- See Also
- Diagnostics
- Bugs

---

---

## lpr

Name  
Command Line  
Description  
Options  
Diagnostics

## lprm

Name  
Command Line  
Description  
Options  
Diagnostics

## net14

Name  
Command Line  
Description  
Usage  
Options

## rsh

Name  
Command Line  
Description  
Options

## setclock

Name  
Command Line  
Description  
Options

---

---

## Overview

This appendix outlines in detail some helpful command programs. Almost all the documentation for each of the commands was taken from their online version written by Sun Microsystems Inc. You can access this same online information by entering `man filename` when you connect to a UNIX machine. For instance, to access information about the finger command, you would enter: `man finger`.

## finger

### Name

The finger command lists information about users.

### Command Line

```
finger [ options ] name...
```

### Description

By default, finger lists information about logged-in users, including their: login name, full name, terminal name (prepended with a ``*` if write-permission is denied), idle time, login time, and location (comment field in `/etc/ttytab` for users logged in locally, hostname for users logged in remotely) if known.

Idle time is minutes if it is a single integer, hours and minutes if a `:` is present, or days and hours if a `d` is present.

When one or more name arguments are given, more detailed information is given for each name specified, whether they are logged in or not. A name may be a first or last name, or an account name. Information is presented in a multi-line format, and includes, in addition to the information mentioned above:

the user's home directory and login shell; the time they logged in; if they are currently logged in, or the time they last logged in if they are not; the terminal or host from which they logged in; and, if a terminal, the comment field in `/etc/ttytab` for that terminal; the last time they received mail; the last time they read their mail; any plan contained in the file `.plan` in the user's home directory; and any project on which they are working, described in the file `.project` (also in that directory)

If a name argument contains an at-sign, ``@'`, then a connection is attempted to the machine named after the at-sign, and the remote finger daemon is queried. The data returned by that daemon is printed. If a long format printout is to be produced, the `lW` option is passed to the remote finger daemon.

---

---

## Options

**-m**

Matches arguments only on user name (not first or last name).

**-l**

Forces long output format.

**-s**

Forces short output format.

**-q**

Forces quick output format, which is similar to short format except that only the login name, terminal, and login time are printed.

**-i**

Forces "idle" output format, which is similar to short format except that only the login name, terminal, login time, and idle time are printed.

**-b**

Suppresses printing the user's home directory and shell in a long format printout.

**-f**

Suppresses printing the header that is normally printed in a non-long format printout.

**-w**

Suppresses printing the full name in a short format printout.

**-h**

Suppresses printing of the .project file in a long format printout.

**-p**

Suppresses printing of the .plan file in a long format printout.

---

---

# ftp

## Name

ftp - file transfer program

## Command Line

```
ftp [ -dginvrsmhfe ] [ hostname ]
```

## Description

ftp is the user interface to the ARPANET standard File Transfer Protocol (FTP). ftp transfers files to and from a remote network site.

The client host with which ftp is to communicate may be specified on the command line. If this is done, ftp immediately attempts to establish a connection to an FTP server on that host; otherwise, ftp enters its command interpreter and awaits instructions from the user. When ftp is awaiting commands from the user, it displays the prompt `ftp>'.

## Options

You may specify the options at the command line or to the command interpreter.

### **-d**

Enables debugging.

### **-e <filename>**

Same as -f except prompts you for login name and password.

### **-f <filename>**

Non-interactive FTP session. Executes commands in <filename>.

### **-g**

Disables filename globbing.

### **-h <filename>**

Gives location of the config.tel file.

### **-i**

Turns off interactive prompting during multiple file transfers.

### **-m**

Uses built-in "more" command.

---

---

**-n**

Does not attempt auto-login upon initial connection. If autologin is enabled, ftp checks the .netrc file in the user's home directory for an entry describing an account on the remote machine. If no entry exists, ftp will prompt for the login name of the account on the remote machine (the default is the login name on the local machine), and, if necessary, prompts for a password and an account with which to login.

**-r**

Disables output redirection.

**-s**

Disables slash flipping.

**-v**

Shows all responses from the remote server, as well as reports on data transfer statistics. This is turned on by default if ftp is running interactively with its input coming from the user's terminal.

## Commands

**! [ command ]**

Runs command as a shell command on the local machine. If no command is given, invoke an interactive shell.

**account [ passwd ]**

Supplies a supplemental password required by a remote system for access to resources once a login has been successfully completed. If no argument is included, the user will be prompted for an account password in a non-echoing input mode.

**ascii**

Sets the representation type to network ASCII. This is the default type.

**bell**

Sounds a bell after each file transfer command is completed.

**binary**

Sets the representation type to image.

---

---

**bye**

Terminates the FTP session with the remote server and exits ftp. An EOF will also terminate the session and exit.

**cd remote-directory**

Changes the working directory on the remote machine to remote-directory.

**close**

Terminates the FTP session with the remote server, and returns to the command interpreter. Any defined macros are erased.

**delete remote-file**

Deletes the file remote-file on the remote machine.

**debug [ debug-value ]**

Toggles debugging mode. If an optional debug-value is specified it is used to set the debugging level. When debugging is on, ftp prints each command sent to the remote machine, preceded by the string `-->`.

**dir [ remote-directory ] [ local-file ]**

Prints a listing of the directory contents in the directory, remote-directory, and, optionally, places the output in local-file. If no directory is specified, the current working directory on the remote machine is used. If no local file is specified, or local-file is `-', output is sent to the terminal.

**get remote-file [ local-file ]**

Retrieves the remote-file and stores it on the local machine. If the local file name is not specified, it is given the same name it has on the remote machine, subject to alteration by the current case, ntrans, and nmap settings. The current settings for representation type, file structure, and transfer mode are used while transferring the file.

**glob**

Toggles filename expansion, or globbing, for mdelete, mget and mput. If globbing is turned off, filenames are taken literally.

Globbing for mput is done as in csh(1). For mdelete and mget, each remote file name is expanded separately on the remote machine, and the lists are not merged.

---

---

Expansion of a directory name is likely to be radically different from expansion of the name of an ordinary file: the exact result depends on the remote operating system and FTP server, and can be previewed by doing ``mls remote-files -'`.

`mget` and `mput` are not meant to transfer entire directory subtrees of files. You can do this by transferring a `tar(1)` archive of the subtree (using a representation type of image as set by the binary command).

### **hash**

Toggles hash-sign (#) printing for each data block transferred. The size of a data block is 1024 bytes.

### **help [ command ]**

Prints an informative message about the meaning of the command. If no argument is given, `ftp` prints a list of the known commands.

### **lcd [ directory ]**

Changes the working directory on the local machine. If no directory is specified, the user's home directory is used.

### **ls [ remote-directory ] [ local-file ]**

Prints an abbreviated listing of the contents of a directory on the remote machine. If `remote-directory` is left unspecified, the current working directory is used. If no `local-file` is specified, or if `local-file` is ``-'`, the output is sent to the terminal.

### **lls**

Works the same as the `ls` command except the local machine is used.

### **mdir remote-files local-file**

Like `dir`, except multiple remote files may be specified. If interactive prompting is on, `ftp` will prompt the user to verify that the last argument is indeed the target local file for receiving `mdir` output.

### **mget remote-files**

---

---

Expands the remote-files on the remote machine and does a get for each file name thus produced. See glob for details on the filename expansion. Resulting file names will then be processed according to case, ntrans, and nmap settings. Files are transferred into the local working directory, which can be changed with `lcd directory`; new local directories can be created with `! mkdir directory`.

**mkdir directory-name**

Makes a directory on the remote machine.

**mls remote-files local-file**

Like ls(1V), except multiple remote files may be specified. If interactive prompting is on, ftp will prompt the user to verify that the last argument is indeed the target local file for receiving mls output.

**mode [ mode-name ]**

Sets the transfer mode to mode-name. The only valid mode-name is stream, which corresponds to the default stream mode.

**more**

Toggles the more mode status. When more mode is on, this command breaks long directories into pages and prompts between displaying them.

**open host [ port ]**

Establishes a connection to the specified host FTP server. An optional port number may be supplied, in which case, ftp will attempt to contact an FTP server at that port. If the auto-login option is on (default), ftp will also attempt to automatically log the user in to the FTP server (see below).

**prompt**

Toggles interactive prompting. Interactive prompting occurs during multiple file transfers to allow the user to selectively retrieve or store files. By default, prompting is turned on. If prompting is turned off, any mget or mput will transfer all files, and any mdelete will delete all files.

**put local-file [ remote-file]**

Stores a local file on the remote machine. If remote-file is left unspecified, the local file name is used after processing according to any ntrans or nmap settings in naming the remote file. File transfer uses the current settings for representation type, file structure, and transfer mode.

---

---

**pwd**

Prints the name of the current working directory on the remote machine.

**quit**

Performs the same function as bye.

**quote arg1 arg2 ...**

Sends the arguments specified, verbatim, to the remote FTP server. A single FTP reply code is expected in return.

**recv remote-file [ local-file]**

Performs the same function as get.

**remotehelp [ command-name ]**

Requests help from the remote FTP server. If a command-name is specified it is supplied to the server as well.

**rename from to**

Renames the file "from" on the remote machine to "to."

**rmdir directory-name**

Deletes a directory on the remote machine.

**send local-file [ remote-file ]**

Performs the same function as put.

**sendport**

Toggles the use of PORT commands. By default, ftp will attempt to use a PORT command when establishing a connection for each data transfer. The use of PORT commands can prevent delays when performing multiple file transfers. If the PORT command fails, ftp will use the default data port. When the use of PORT commands is disabled, no attempt will be made to use PORT commands for each data transfer. This is useful when connected to certain FTP implementations that ignore PORT commands but incorrectly indicate they have been accepted.

**slashflip**

Toggles slashflipping command.

---

---

**status**

Shows the current status of ftp.

**struct [ struct-name ]**

Sets the file structure to struct-name. The only valid struct-name is file, which corresponds to the default file structure.

**type [ type-name ]**

Sets the representation type to type-name. The valid type-names are ascii for network ASCII, binary or image for image, and tenex for local byte size with a byte size of 8 (used to talk to TENEX machines). If no type is specified, the current type is printed. The default type is network ASCII.

**user user-name [ password ] [ account ]**

Identifies yourself to the remote FTP server. If the password is not specified and the server requires it, ftp will prompt the user for it (after disabling local echo). If an account field is not specified, and the FTP server requires it, the user will be prompted for it. If an account field is specified, an account command will be relayed to the remote server after the login sequence is completed if the remote server did not require it for logging in. Unless ftp is invoked with auto-login disabled, this process is done automatically on initial connection to the FTP server.

**verbose**

Toggles verbose mode. In verbose mode, all responses from the FTP server are displayed to the user. In addition, if verbose mode is on, when a file transfer completes, statistics regarding the efficiency of the transfer are reported. By default, verbose mode is on if ftp's commands are coming from a terminal, and off otherwise.

**? [ command ]**

Performs the same function as help.

Command arguments which have embedded spaces may be quoted with quote (") marks.

If any command argument which is not indicated as being optional is not specified, ftp will prompt for that argument.

---

---

## Aborting a File Transfer

To abort a file transfer, use the terminal interrupt key (usually CTRL-C). Sending transfers will be immediately halted. Receiving transfers will be halted by sending a ftp protocol ABOR command to the remote server, and discarding any further data received. The speed at which this is accomplished depends upon the remote server's support for ABOR processing. If the remote server does not support the ABOR command, an ftp> prompt will not appear until the remote server has completed sending the requested file.

The terminal interrupt key sequence will be ignored when ftp has completed any local processing and is awaiting a reply from the remote server. A long delay in this mode may result from the ABOR processing described above, or from unexpected behavior by the remote server, including violations of the ftp protocol. If the delay results from unexpected remote server behavior, the local ftp program must be killed by hand.

## File Naming Conventions

Local files specified as arguments to ftp commands are processed according to the following rules.

1. If globbing is enabled, local file names are expanded according to the rules used in the csh(1); see the glob command. If the ftp command expects a single local file (for example, put), only the first filename generated by the globbing operation is used.
2. For mget commands and get commands with unspecified local file names, the local filename is the remote filename, which may be altered by a case, ntrans, or nmap setting. The resulting filename may then be altered if runique is on.
3. For mput commands and put commands with unspecified remote file names, the remote filename is the local filename, which may be altered by a ntrans or nmap setting. The resulting filename may then be altered by the remote server if sunique is on.

## File Transfer Parameters

The FTP specification specifies many parameters which may affect a file transfer.

The representation type may be one of network ASCII, EBCDIC, image, or local byte size with a specified byte size (for PDP-10's and PDP-20's mostly). The network ASCII and EBCDIC(rq) types have a further subtype which specifies whether vertical format control (NEWLINE characters, form feeds, etc.) are to be passed through (non-print), provided in TELNET format (TELNET format controls), or provided in ASA (FORTRAN) (carriage control (ASA)) format. ftp supports the network ASCII (subtype non-print only) and image types, plus local byte size with a byte size of 8 for communicating with TENEX machines.

---

---

The file structure may be one of file (no record structure), record, or page. ftp supports only the default value, which is file.

The transfer mode may be one of stream, block, or compressed. ftp supports only the default value, which is stream.

## lpq

### Name

lpq - display the queue of printer jobs.

### Command Line

```
lpq [ -Pprinter ] [ -l ] [ + [ interval ] ] [ job# ... ] [ username ... ]
```

### Description

lpq displays the contents of a printer queue. It reports the status of jobs specified by job#, or all jobs owned by the user specified by username. lpq reports on all jobs in the default printer queue when invoked with no arguments.

For each print job in the queue, lpq reports the user's name, current position, the names of input files comprising the job, the job number (by which it is referred to when using lprm(1)) and the total size in bytes. Normally, only as much information as will fit on one line is displayed. Jobs are normally queued on a first-in-first-out basis. Filenames comprising a job may be unavailable, such as when lpr is used at the end of a pipeline; in such cases the filename field indicates ``(standard input)"

If lpq warns that there is no daemon present (that is, due to some malfunction), the lpc(8) command can be used to restart a printer daemon.

### Options

#### **-P printer**

Displays information about the queue for the specified printer. In the absence of the -P option, the queue to the printer specified by the PRINTER variable in the environment is used. If the PRINTER variable isn't set, the queue for the default printer is used.

#### **-l**

Displays queue information in long format; includes the name of the host from which the job originated.

#### **+*[interval]***

Displays the spool queue periodically until it empties. This option clears the terminal screen before reporting on the queue. If an interval is supplied, lpq sleeps that number of seconds in between reports.

---

---

## Diagnostics

### **Printer Is Ready and Printing**

The lpq program checks to see if there is a printer daemon. If the daemon is hung, the super-user can abort the current daemon and start a new one using lpc(8). Under some circumstances, lpq reports that a printer is ready and printing when the daemon is, in fact, hung.

### **Waiting for Printer to Become Ready (offline ?)**

The daemon could not open the printer device. The printer may be turned off-line. This message can also occur if a printer is out of paper, the paper is jammed, and so on. Another possible cause is that a process, such as an output filter, has exclusive use of the device. The only recourse in this case is to kill the offending process and restart the printer with lpc.

### **Waiting for Host to Come Up**

A daemon is trying to connect to the remote machine named host, in order to send the files in the local queue. If the remote machine is up, lpd on the remote machine is probably dead or hung and should be restarted using lpc.

### **Sending to Host**

The files are being transferred to the remote host, or else the local daemon has hung while trying to transfer the files.

### **Warning: Printer Is Down**

The printer has been marked as being unavailable with lpc.

### **Warning: No Daemon Present**

The lpq process overseeing the spooling queue, as indicated in the ``lock" file in that directory, does not exist. This normally occurs only when the daemon has unexpectedly died. Check the printer's error log for a diagnostic from the deceased process; you can restart the printer daemon with lpc.

---

---

# lpr

## Name

lpr - send a job to the printer

## Command Line

```
lpr [ -Pprinter ] [ -#copies ] [ -Cclass ] [ -Jjob ] [ -Ttitle ] [ -i  
[ indent ] ] [ -1234font ] [ -wcols ] [ -B ] [ -r ] [ -m ] [ -h ] [ -s ]  
[ -filter-option ] [ filename ... ]
```

## Description

lpr forwards printer jobs to a spooling area for subsequent printing as facilities become available. Each printer job consists of copies of (or, with -s, symbolic links to) each filename you specify. The spool area is managed by the line printer daemon, lpd(8). lpr reads from the standard input if no files are specified.

## Options

### **-Pprinter**

Sends output to the named printer. Otherwise send output to the printer named in the PRINTER environment variable, or to the default printer, lp.

### **-#copies**

Produces the number of copies indicated for each named file. For example:

```
example% lpr -#3 index.c lookup.c
```

produces three copies of index.c, followed by three copies of lookup.c. On the other hand,

```
example% cat index.c lookup.c | lpr
```

generates three copies of the concatenation of the files.

### **-Cclass**

Prints class as the job classification on the burst page. For example,

```
example% lpr -C Operations new.index.c
```

replaces the system name (the name returned by hostname) with "Operations" on the burst page, and prints the file new.index.c.

### **-Jjob**

Prints job as the job name on the burst page. Normally, lpr uses the first file's name.

---

---

**-T title**

Uses title instead of the file name for the title used by pr(1V).

**-i[indent]**

Indents output indent spaces. Eight spaces is the default.

-1 font

-2 font

-3 font

-4 font

Mounts the specified font on font position 1, 2, 3 or 4. The daemon will construct a .railmag file in the spool directory that indicates the mount by referencing /usr/lib/vfont/font.

**-wcols**

Uses cols as the page width for pr.

**-r**

Removes the file upon completion of spooling, or upon completion of printing with the -s option.

**-m**

Sends mail upon completion.

**-h**

Suppresses printing the burst page.

**-s**

Creates a symbolic link from the spool area to the data files rather than trying to copy them (so large files can be printed). This means the data files should not be modified or removed until they have been printed. In the absence of this option, files larger than 1 Megabyte in length are truncated. Note: the -s option only works on the local host (files sent to remote printer hosts are copied anyway), and only with named data files - it doesn't work if lpr is at the end of a pipeline.

---

---

**filter-option**

The following single letter options notify the line printer spooler that the files are not standard text files. The spooling daemon will use the appropriate filters to print the data accordingly.

**-p**

Uses pr to format the files (lpr-p is very much like pr | lpr).

**-l**

Prints control characters and suppresses page breaks.

**-t**

The files contain troff(1) (cat phototypesetter) binary data.

**-n**

The files contain data from ditroff (device independent troff).

**-d**

The files contain data from tex (DVI format from Stanford).

**-g**

The files contain standard plot data as produced by the plot(3X) routines (see also plot(1G) for the filters used by the printer spooler).

**-v**

The files contain a raster image, see rasterfile (5).

**-c**

The files contain data produced by cifplot.

**-f**

Interprets the first character of each line as a standard FORTRAN carriage control character. If no filter-option is given (and the printer can interpret PostScript), the string `%' as the first two characters of a file indicates that it contains PostScript commands.

---

---

## Diagnostics

### **lpr: copy file is too large**

A file is determined to be too "large" to print by copying into the spool area. Use the `-s` option as defined above to make a symbolic link to the file instead of copying it. A too-large file is approximately 1 Megabyte. `lpr` truncates the file, and prints as much of it as it can.

### **lpr: printer: unknown printer**

The printer was not found in the `printcap` database. Usually this is a typing mistake; however, it may indicate a missing or incorrect entry in the `/etc/printcap` file.

### **lpr: printer: jobs queued, but cannot start daemon.**

The connection to `lpd` on the local machine failed. This usually means the printer server started at boot time has died or is hung. Check the local socket `/dev/printer` to be sure it still exists (if it does not exist, there is no `lpd` process running).

### **lpr: printer: printer queue is disabled**

This means the queue was turned off with

```
example% /usr/etc/lpc disable printer
```

to prevent `lpr` from putting files in the queue. This is normally done by the system manager when a printer is going to be down for a long time. The printer can be turned back on by a super-user with `lpc`.

If a connection to `lpd` on the local machine cannot be made `lpr` will say that the daemon cannot be started. Diagnostics may be printed in the daemon's log file regarding missing spool files by `lpd`

## lprm

### Name

`lprm` - remove jobs from the printer queue

### Command Line

```
lprm [ -Pprinter ] [ - ] [ job # ... ] [ username ... ]
```

### Description

`lprm` removes a job or jobs from a printer's spooling queue. Since the spool directory is protected from users, using `lprm` is normally the only method by which a user can remove a job.

Without any arguments, `lprm` deletes the job that is currently active, provided that the user who invoked `lprm` owns that job.

---

---

You can remove a specific job by supplying its job number as an argument, which you can obtain using `lpq(1)`. For example:

```
example% lpq -Phost
host is ready and printing
Rank Owner Job Files Total Size
active wendy 385 standard input 35501 bytes
example% lprm -Phost 385
```

`lprm` reports the names of any files it removes, and is silent if there are no applicable jobs to remove.

`lprm` kills the active printer daemon, if necessary, before removing spooled jobs; it restarts the daemon when through.

## Options

### **-Pprinter**

Specifies the queue associated with a specific printer. Otherwise the value of the `PRINTER` variable in the environment is used. If this variable is unset, the queue for the default printer is used.

Removes all jobs owned by you. If invoked by the super-user, all jobs in the spool are removed. (Job ownership is determined by the user's login name and host name on the machine where the `lpr` command was invoked).

## Diagnostics

### **lpr: printer: jobs queued, but cannot start daemon.**

The connection to `lpd` on the local machine failed. This usually means the printer server started at boot time has died or is hung. If it is hung, the master `lpd(8)` daemon may have to be killed and a new one started.

## net14

### Name

`net14` is a utility that redirects serial output onto a TCP/IP network.

### Command Line

```
net14 -h <filename> <program-name> [program options]
```

### Full Description

`net14` is a utility to allow programs which use interrupt 14h to output serial information to output that information onto a TCP/IP network. Thus, you could use this utility to re-direct the output from MS-Kermit onto a TCP/IP network through an ethernet card, while still retaining the MS-Kermit user interface.

---

---

## Usage

The net14 program acts as a shell in which you can execute another program. net14 first initializes the network and reads in the config.tel file, then transfers control to the program specified on the command line. For example, the following line starts net14, specifies a location for the config.tel file, and fires up MS-Kermit:

```
net14 -h c:\ncsa\config.tel kermit
```

To pass parameters to the program which net14 transfers control to, place them after the filename of the program.

## Options

### **-h <filename>**

Specifies the location of the config.tel file to read in the network options.

### **<program-name>**

The program which net14 executes after initializing the network. net14 searches for the program in the path specified in the MS-DOS PATH environment variable.

### **[program options]**

This program passes options to the program which net14 executes.

## rsh

### Name

rsh - remote shell

### Command Line

```
rsh [-l username] [-n] hostname command
```

```
rsh hostname [-l username] [-n] command
```

```
hostname [-l username] [-n] command
```

### Description

rsh connects to the specified hostname and executes the specified command. rsh copies its standard input to the remote command, the standard output of the remote command to its standard output, and the standard error of the remote command to its standard error. Interrupt, quit and terminate signals are propagated to the remote command; rsh normally terminates when the remote command does.

---

---

If you omit the command, instead of executing a single command, rsh logs you in on the remote host using rlogin(1C). Shell metacharacters which are not quoted are interpreted on the local machine, while quoted metacharacters are interpreted on the remote machine.

Hostnames are given in the hosts database, which may be contained in the /etc/hosts file, the Yellow Pages hosts database, the Internet domain name database, or some combination of the three. Each host has one official name (the first name in the database entry) and optionally one or more nicknames. Official hostnames or nicknames may be given as hostname.

If the name of the file from which rsh is executed is anything other than "rsh," rsh takes this name as its hostname argument. This allows you to create a symbolic link to rsh in the name of a host which, when executed, will invoke a remote shell on that host. The /usr/hosts directory is provided to be populated with symbolic links in the names of commonly used hosts. By including /usr/hosts in your shell's search path, you can run rsh by typing hostname to your shell.

Each remote machine may have a file named /etc/hosts.equiv that contains a list of trusted hostnames with which it shares usernames. Anyone with identical usernames on the local and remote machines may rsh from the machines listed in the remote machine's /etc/hosts file. Each user can set up a similar private equivalence list using the file .rhosts in their home directories. Each line in this file contains (at least) a hostname, a separating space, and a username. The entry permits the user named username who is logged into hostname to use rsh to access the remote machine as the remote user. If the name of the local host is not found in the /etc/hosts.equiv file on the remote machine, and the local username and hostname are not found in the remote user's .rhosts file, then the access is denied. The hostnames listed in the /etc/hosts.equiv and .rhosts files must be the official hostnames listed in the hosts database. You cannot use nicknames in either of these files. Unless the command argument is omitted, rsh will not prompt you for a password if the remote machine denies you access.

## Options

### **-l username**

Uses the username as the remote username instead of your local username. Without this option, the remote username and your local username are identical.

### **-n**

Redirects the input of rsh to /dev/null. Occasionally, you must use this option to avoid unfortunate interactions between rsh and the shell which invokes it. For example, if you are running rsh and invoke a rsh in the background without redirecting its input away from the terminal, it will block even if the remote command doesn't post any reads. The -n option prevents this situation.

---

---

Your entry in the file `/etc/passwd` on the remote system determines the type of remote shell (sh, rsh, or other) your local system will use.

## setclock

### Name

setclock sets the local clock to one on a network server.

### Command Line

```
setclock -h <filename> hostname
```

### Description

The utility, setclock, sets the PC's local clock from the network's clock server.

### Options

**-h <filename>**

Gives the location and the name of the config.tel file.

**hostname**

Gives the name of the host from which you'll get the time.

---