

# 4

## Network Clients and Utilities

<i>Page</i>	<i>Processes</i>	<i>Page</i>	<i>Processes by Function</i>
76	finger	<b>76</b>	<b>Informational Utilities</b>
87	ftp	76	finger—Retrieve information about remote users.
92	mail	77	netstat—Display network information.
94	mail (UCB Version)	79	nslookup—Perform a DNS lookup.
77	netstat	81	ping—Check if a host is alive.
79	nslookup	82	rpcinfo—Show remote procedure call information.
81	ping	83	rup—Remote uptime utility.
90	rcp	84	ruptime—Check remote system information.
98	rlogin	84	rusers—Check remote user information.
82	rpcinfo	85	traceroute—Trace the path of packets between hosts.
99	rsh	86	whois—Look up domain registration information.
83	rup	<b>87</b>	<b>File Transfer</b>
84	ruptime	87	ftp—File Transfer Protocol utility.
84	rusers	90	rcp—Remote copy.
97	rwall	91	rwho—Remote user information.
91	rwho	<b>92</b>	<b>Communications</b>
97	talk	92	mail—Send or read email.
100	telnet	94	mail (UCB version).
85	traceroute	97	rwall—Send message to all users on network.
86	whois	97	talk—Chat with another user.
98	write	98	write—Send message to a local user.
		<b>98</b>	<b>Remote Shells and Login</b>
		98	rlogin—Remote login.
		99	rsh—Remote shell.
		100	telnet—Log in to a remote system.

## Informational Utilities

### finger

```
finger /usr/bin/finger [options] [user]
       /usr/bin/finger [options] [user@host]
       /usr/bin/finger [options] [user@host@host@host.... etc.]
```

---

Retrieve information about users from the current host or remote servers. The remote server must be running a finger daemon (`in.fingerd`).

---

Option	Description
-b	Do not show home directories and shells in the information given.
-f	Do not print headers.
-h	Do not show <code>.project</code> files.
-I	Idle format. Output includes username, terminal, login time, and idle time.
-l	Long format. Output includes all information.
-m	Match <code>user</code> with username and not the full name of the user.
-p	Do not show <code>.plan</code> files.
-q	Quick format. Output includes username, terminal, and login time.
-s	Short format.
-w	Short format without printing full name.

#### Files

File	Description
<code>.project</code>	Text file that can optionally be kept in the user's home directory. If present, it is printed in the <code>Project:</code> field of the finger output.
<code>.plan</code>	Text file that can optionally be kept in the user's home directory. If present, it is printed at the end of the finger information for the user.

## Finger Forwarding

*Finger forwarding* is when a `finger` request is sent to a remote host and then “forwarded” to another host, where it is processed and the resulting information is sent back to the original sender of the request. Consider the following example:

```
finger smithj@host1.com@host2.com@host3.com
```

The `finger` request will go from the local host to `host3.com`.

`host3.com` will forward the request to `host2.com`, `host2.com` will forward the request to `host1.com`, and `host1.com` will send the `finger` information (if available) back to the local host.

In each case, the request will be logged on the remote system as coming from the forwarding host. `host1.com` will log the request as being from the local host. It is for this reason that some system administrators disable `finger` forwarding on their servers. This is a recommended practice that can potentially make your entire network more secure. See `in.fingerd` for details.

`Finger` forwarding is not supported on all servers. The specifications for `finger` forwarding are defined in RFC 1288.

### Example

```
finger
```

### Description

To retrieve `finger` information for all users on the local host.

```
finger user
```

To retrieve `finger` information for a single user on the local system.

```
finger @remotehost.com
```

To retrieve `finger` information for all users currently logged into a remote host.

```
finger user@remotehost.com
```

To retrieve `finger` information for a single user on a remote host.

► **See Also** `rwho` (91), `whois` (86)

## netstat

```
netstat [options] [interval]
```

netstat

---

Show current network information for the local host. This command can be used to show all network connections to the local system, routing information, and network interface information.

---

Option	Description
[ <i>interval</i> ]	An interval can be specified, causing <code>netstat</code> to report the network status every <i>interval</i> seconds. This interval is optional. If omitted, <code>netstat</code> will run once.
-a	All socket and routing table states and entries are shown. If this option is omitted, daemon processes will not be shown and only limited routing table information will be given.
-f <i>fam</i>	Only show reports for those that are of the specified family. The family can be either of the following: <ul style="list-style-type: none"> <li>■ <code>inet</code>: AF_INET address family</li> <li>■ <code>unix</code>: AF_UNIX address family</li> </ul>
-g	Multicast group memberships for all interfaces are displayed.
-I	Display TCP/IP interface state.
-I <i>interface</i>	The current state of the network interface specified by <i>interface</i> is shown.
-m	Display statistics for STREAMS.
-M	Multicast routing tables are shown. Can be used with the <code>-s</code> option to give summary statistics using the multicast routing tables.
-n	Do not resolve addresses. This option will cause any hosts to be reported by IP address rather than by hostname. Using this usually greatly increases the speed of the <code>netstat</code> command.
-p	Display address resolution tables (ARP tables).
-p <i>protocol</i>	Only show network statistics for sockets using the protocol specified by <i>protocol</i> . This can be <code>tcp</code> or <code>udp</code> .
-r	Display routing tables.
-s	Give statistics sorted by protocol, such as <code>tcp</code> or <code>udp</code> . If used with <code>-M</code> , multicast routing statistics are used instead.
-v	Verbose. Reports are given in more detail.

### TCP Socket States

State	Description
CLOSED	Closed. The socket has been closed and is not in use.
CLOSE_WAIT	Wait for close. The remote side has closed the connection and the local host is waiting for the socket to close.

*continues >>*

&gt;&gt;continued

State	Description
CLOSING	Closing. Socket has been closed and the remote connection has been shut down. The socket is waiting for acknowledgment.
ESTABLISHED	Connection established. A TCP/IP connection is established and working.
FIN_WAIT_1	Finish wait 1. The socket has been successfully closed and waiting to shut down the connection.
FIN_WAIT_2	Finish wait 2. Socket has been closed and is waiting for remote side to shut down connection.
LAST_ACK	Last acknowledgment. The remote side of the connection has been shut down and closed. Waiting for acknowledgment.
LISTEN	Listening. The socket is currently listening. Programs such as servers and daemons will usually spawn processes that open sockets to listen.
SYN_SENT	Synchronization sent. The socket is attempting to establish a connection with a remote host.
SYN_RECEIVED	Synchronization received. A response has been received after sending a synchronization request. The connection is being made.
TIME_WAIT	Wait after close. Socket has been closed, waiting for remote shutdown retransmission.

**Example****Description**

<code>netstat</code>	Check what remote hosts are connected to the local system.
<code>netstat -n</code>	Check what remote hosts are connected to the local system, but leave addresses as IP numbers (much faster).

► **See Also** `rpcinfo` (82)

## nslookup

`/usr/sbin/nslookup [options] [host] [DNS server]`

nslookup

Utility to query an Internet domain name server to resolve an IP address to a name or a name to an IP address. By default, `nslookup` uses the server in `/etc/resolv.conf` as the DNS server. It can be used in two general ways:

- Hostname specified on the command line.
- Run without hostname specified to use interactive mode.

## Command-Line Options

Option	Description
<i>-option</i>	Set or change lookup state information. For example:  -retry=5  See the later section “State Keywords” for a list.
DNS <i>server</i>	Specify a DNS server. If no DNS server is given, the one listed in <i>/etc/resolv.conf</i> is used.
host	Host can be either a domain name or an IP address. If a host is specified on the command line, the results are returned and interactive mode is not used.

## Interactive Mode Commands

Command	Description
<b>CTRL+D</b>	Exit.
exit	Exit.
finger[ <i>name</i> ]	Finger the last host that was successfully looked up.
help	Show a command summary. Using ? is the same as using help.
host[ <i>server</i> ]	Look up host using <i>server</i> as the DNS server. If <i>server</i> is not given, the DNS server listed in <i>/etc/resolv.conf</i> is used.
ls [-ah]	List all domain name service information for the domain: <ul style="list-style-type: none"> <li>■ -a: Show all aliases as well.</li> <li>■ -h: Show host information including CPU and OS information.</li> </ul>
root	Change the default server to <i>ns.nic.ddn.mil</i> .
set	Set a keyword value. Keyword definitions must be of one of the following forms:  set <i>keyword</i>  set <i>keyword</i> = <i>value</i>  See the subsequent “State Keywords” section for a list of all keywords that can be set.
server <i>domain</i>	Change the default server to the domain specified by <i>domain</i> .
view <i>file</i>	Use more to view the given <i>file</i> .

## State Keywords

Keyword	Description
all	Show all set keywords and their values.
debug or nodebug	Turn debugging on or off. The default is <code>nodebug</code> .
defname or nodename	If <code>defname</code> is set, the default domain name is appended to every hostname that is looked up. The default is <code>nodename</code> .
domain = <i>file</i>	Change the default domain to <i>file</i> . If no file is given, the domain given in <code>/etc/resolv.conf</code> is used.
querytype = <i>type</i>	Set query type (specified in RFC 833): <ul style="list-style-type: none"> <li>■ A: Host Internet address (default)</li> <li>■ CNAME: Canonical name</li> <li>■ HINFO: Host CPU and OS type</li> <li>■ MD: Mail destination</li> <li>■ MX: Mail exchanger</li> <li>■ MB: Mailbox domain name</li> <li>■ MG: Mail group member</li> <li>■ MINFO: Mailbox or mail list information</li> </ul>
recurse or norecurse	If <code>recurse</code> is set and the current server does not have the information, it will check other servers. The default is <code>recurse</code> .
retry= <i>n</i>	Retry the request <i>n</i> times if the initial request fails.
root= <i>host</i>	Change the root server (default is <code>ns.nic.ddn.mil</code> ) to <i>host</i> .
timeout= <i>seconds</i>	Set the timeout for a query to the number of seconds specified by <i>seconds</i> .
vc or novc	Force the use of a virtual circuit. The default is <code>novc</code> .

►See Also `whois` (86)

## ping

```
/usr/sbin/ping [options] [host] [timeout]
```

ping

Uses the Internet Control Message Protocol (ICMP) to check whether a remote host is alive on the network.

Local host→ECHO\_REQUEST→ Remote host

Local host←ECHO\_RESPONSE← Remote host

Command	Description
-I <i>interface</i>	Use the interface specified by <i>interface</i> for outgoing packets.
-I <i>n</i>	Interval. Wait <i>n</i> seconds between pings.
-l	Loose source route. The packet will find a route to the destination host. Using this option, the reply packet will discover a route back to the originating host (not necessarily the same as the original route). This can be used to diagnose possible routing problems between two hosts on a network.
-L	No loopback of multicast packets. Do not copy multicast packets to members of the host group of the interface.
-n	Numerical addresses. Show IP numbers for hosts rather than DNS names.
-r	Skip routing tables. Packet is sent directly to host. This only works if the remote host is on the same network segment.
-R	Route recording. The route of the packet is stored in the IP header.
-t <i>lifetime</i>	Time to live for packets is set to <i>lifetime</i> . The default is one hop.
-v	Verbose output. List all ICMP packets received.

### Example

Test connectivity as seen by NFS packets (using 8192 byte packets). Show degradation in passing through alternative packet-sized routing equipment.

```
ping -sv hostname 8192
```

►See Also `whois` (86)

## rpcinfo

```
rpcinfo /usr/bin/rpcinfo [options] [host] [prog] [version]
```

Use RPC calls to report RPC information. RPC information for both local and remote hosts can be reported. In general, `rpcinfo` can be used in three ways:

- List all registered RPC services on a host.
- List all `rpcbind` version 2 registered RPC services on a host.
- Make a call (procedure 0) to a specific program on a host.

Option	Description
-a <i>addr</i>	Use the address specified by <i>addr</i> as the universal address for the service. Used to ping a remote service.
-b	Broadcast procedure 0 of the service listed and report all responses from the network that are received.
-d	Delete a service registration for the specified program and version number.
-l	List entries matching the specified program and version number on a host.
-m	Show summary statistics for rcpbind on a remote host. Statistics for rcpbind versions 2, 3, and 4 are shown.
-p	Probe. The host is probed using version 2 of the rcpbind protocol. If a host is specified, that host is probed. If no host is specified, the local host is probed.
-s	Short list. Give a short list of all RPC programs registered on the specified remote host, or on the local host if no remote host is specified.
-T <i>transport</i>	Require the RPC service to be on the transport specified by <i>transport</i> . If this option is omitted, the transport in the environment variable NETPATH is used. If NETPATH is not set, the netconfig database is used to determine the transport.

**Example**

rpcinfo

**Description**

Check what RPC services are registered on the local host.

rpcinfo *remotehost*

Check what RPC services are registered on a remote host.

rpcinfo -p *remotehost*

Check what RPC services are registered on a remote host, showing only version 2 programs.

**rup**

```
/usr/sbin/rup [options] [host]
```

rup

---

Similar to the uptime command, but displays information for remote hosts. If no host is specified, the request is broadcast to all machines. Remote systems must be running the rstatd daemon to respond. Note that ruptime uses the in.rwhod daemon, whereas rup uses the rstatd daemon.

---

Option	Description
-h	Sort output by host name.
-l	Sort output by load average.
-t	Sort output by uptime.

►See Also `ruptime` (84)

## ruptime

`ruptime` /usr/bin/ruptime [options]

---

Check information for a remote host. Information given is similar to that of `uptime`. Each machine on the local network is queried and a response must be received within five minutes. The remote server must be running the remote `who` daemon (`in.rwhod`). Note that `ruptime` uses the `in.rwhod` daemon, whereas `rup` uses the `rstatd` daemon.

---

Option	Description
-a	Include all users in the output. If omitted, any user idle for more than an hour will be excluded.
-l	Output sorted by load average.
-r	Output given in reverse order.
-t	Output sorted by uptime.
-u	Output sorted by number of users.

►See Also `rwho` (91), `rusers` (84)

## rusers

`rusers` /usr/bin/rusers [options] [host]

---

A remote version of `who` to find out who is logged in to a remote host. More than one host can be specified on the command line. If no hosts are specified, `rusers` sends out a broadcast for `rusersd` protocol, version 3. This is followed by a broadcast of version 2. Hosts must be running the `ruserd` daemon in order to respond.

---

Option	Description
-a	Report machines even with no users logged in.
-h	Alphabetically list hosts.
-i	Reports are sorted by idle time.

Option	Description
-l	Long report format, giving more detail.
-u	Reports are sorted by the number of users logged in.

►See Also `rwall` (97), `rwho` (91)

## traceroute

```
/usr/sbin/traceroute [options] host [len]
```

traceroute

Trace the route IP packets take from one Internet/intranet host to another host. This utility can use both IPv4 (time to live) and IPv6 (hop limit). The maximum hop limit is set to 30 by default but can be changed by using the `-m` option. `traceroute` traces the route from one host to another host at that moment. Given the dynamic nature of Internet routing today, results may vary from moment to moment.

Option	Description
-A addressfam	Specify the address family for the target host. Arguments can be <code>inet</code> for IPv4 and <code>inet6</code> for IPv6.
-a	For multi-homed hosts, this option will probe all IP addresses.
-c class	Specify the probe packet traffic class as an integer from 0 to 255.
-d	Set the debugging socket option ( <code>SO_DEBUG</code> ).
-F	Set the no-fragment option.
-f firsthop	Override the default first-hop setting of 1, and set the starting time-to-live hop limit to <i>firsthop</i> .
-g gateway	Set up to eight additional gateways to use.
-I	Use ICMP ECHO rather than UDP datagrams.
-i interface	Specify source IP network interface (for IPv4) or the packet-transmitting interface (for IPv6). Can be an integer representing an interface index or an interface name (such as <code>le0</code> ).
-L flow	Set the IPv6 flow label for the packets. Must be an integer between 0 and 1048575.
-l	Show the time-to-live for each packet received.
-m maxhop	Override the default, maximum time-to-live value of 30, and set it to <i>maxhop</i> .

*continues >>*

&gt;&gt;continued

Option	Description
-n	Do not perform DNS lookups on IP addresses.
-P seconds	Specify a pause (in seconds) between probe packets.
-p port	Specify the base UDP port to use when probing. Default value is 33434.
-Q timeouts	Stop the trace after the specified number of timeouts have been received.
-q queries	Specify the number of queries for probing. Default value is 3.
-r	Skip routing tables and send packets directly to host on network. Cannot be used with -g.
-s sourceaddr	Specify the source address when sending packets. The source address must match one of the machine's interfaces.
-t servicetype	Specify the type of service (ToS) for outgoing packets. Must be between 0 and 255. Default value is 0.
-v	Verbose. Show extra information when tracing, including packet size and destination.
-w seconds	Specify the time to wait for a response to a probe packet. Default value is 5.
-x	Skip checksums when using IPv4.

**Example**

traceroute www.microsoft.com

**Description**

Trace packet path from localhost to www.microsoft.com

traceroute -vn helios

Trace packet path from localhost to a machine called helios on your LAN and do not perform DNS lookups, showing verbose output.

►See Also netstat (77), ping (81)

**whois**

whois /usr/bin/whois [options] [string]

---

Check the InterNIC database for a domain record or records. If an exact match is found for the string, the domain record for the domain is printed. If the string matches multiple domains, summaries of the domains are shown. Other NICs can be checked by specifying the host using the -h option.

---

Example	Description
<code>whois somedomain.com</code>	Check if a domain is registered and/or find out contact and billing information for the domain.
<code>whois [string]</code>	Find all domains containing a certain string.

Option	Description
-h	Specify which host to use for lookups to find information at other NICs (optional).

►See Also `finger` (76)

## File Transfers

### ftp

`/usr/bin/ftp [options] [host]`

ftp

File Transfer Protocol (FTP) client to upload and download files over the network. If no host is specified on the command line, an `ftp>` command prompt is given. The remote server must be running an FTP daemon (`in.ftpd`).

Option	Description
-d	Debugging mode.
-g	Do not use filename globbing.
-i	Do not ask for confirmation for each file of a multiple file transfer (non-interactive mode).
-n	Disable auto-login.
-v	Verbose. Show all diagnostics and give a summary of data transfer statistics.

Command	Description
? <i>command</i>	Same as <code>help</code> .
! <i>command</i>	Run <i>command</i> using the shell. If no command is given, the shell is used as the command interpreter until <b>exit</b> is typed.

*continues >>*

&gt;&gt;continued

Command	Description
\$ <i>macro</i>	Run the macro specified by <i>macro</i> . See <i>macdef</i> , later in this table.
account ↳ <i>password</i>	Provide an extra password to the remote system if necessary. If no password is given, the user is prompted for one.
append ↳ <i>file1 file2</i>	Append the local file <i>file1</i> to the remote file, <i>file2</i> .
ascii	Set the transfer mode to ASCII (rather than binary).
bell	Toggle beeping after each file transfer.
binary	Set the transfer mode to binary (rather than ASCII).
bye	Close session and exit.
case	Toggle case mapping. Default is <i>off</i> . <ul style="list-style-type: none"> <li>■ <i>on</i>: All uppercase characters on the remote system are changed to lowercase.</li> <li>■ <i>off</i>: No changes are made to upper- or lowercase characters.</li> </ul>
cd <i>dir</i>	Change directories on the remote host to <i>dir</i> .
cdup	Change the current directory on the remote host to the parent directory.
close	Close FTP session.
cr	Toggle carriage-return (CR) stripping during ASCII mode.
delete <i>file</i>	Delete file on the remote host.
debug	Toggle debugging mode. Default is <i>off</i> .
dir	Give a directory listing of the current remote host working directory. Output is similar to that of <i>ls</i> .
disconnect	Same as <i>close</i> .
get <i>file</i> ↳ <i>filename</i>	Download the file specified by <i>file</i> from the remote host to the local host. If <i>filename</i> is given, that name is used on the local host. If not, the remote name is used.
glob	Toggle globbing for use with <i>mdelete</i> , <i>mget</i> , and <i>mput</i> . Globbing is filename expansion, the same as done by <i>sh</i> .
hash	Toggle hash marks (#). If on, hash marks will usually be printed for every 8,192 bytes transferred. Some systems may use a different hash mark size, in which case it will be specified. Another common hash mark size is 2,048 bytes.

continues &gt;&gt;

&gt;&gt;continued

Command	Description
help <i>command</i>	Give help on <i>command</i> . If no command is specified, a list of commands is displayed.
lcd <i>dir</i>	Change local directory to <i>dir</i> .
ls	Similar to <i>dir</i> , but gives a briefer directory listing. If the <i>-a</i> option is given, all files are listed, including those that begin with a dot ( <i>.</i> ).
macdef <i>macro</i>	Define a macro by the name of <i>macro</i> . Input following <i>macdef</i> will be stored as the macro until a new line is given.
mdelete <i>files</i>	Multiple <i>delete</i> . Delete the file or files given by <i>files</i> .
mdir <i>files</i>	Multiple directory listing. List the files or directories specified.
mget <i>files</i>	Multiple <i>get</i> . Download all the files specified.
mkdir <i>dir</i>	Make directory. Create a directory on the remote system called <i>dir</i> .
mls <i>files</i>	Multiple <i>ls</i> . Same as <i>ls</i> , but more than one file or directory can be given.
mput <i>files</i>	Multiple <i>put</i> . Upload all of the files specified.
open <i>host port</i>	Open a connection with host on port. If no port is given, 21/tcp is used.
prompt	Toggle interactive prompting. If this is off, no prompt will be given between file transfers when using <i>mput</i> or <i>mget</i> .
proxy <i>command</i>	Run <i>command</i> on a secondary server. This can be used to transfer files between two remote servers.
put <i>file</i> ↳ <i>filename</i>	Upload <i>file</i> to the remote host. If <i>filename</i> is given, it is renamed to <i>filename</i> on the remote host. If no <i>filename</i> is given, the original filename is used on the remote server.
pwd	Print the current working directory.
quit	Same as <i>bye</i> .
quote <i>args</i>	Send the specified file directly to the remote FTP server.
recv <i>file</i>	Same as <i>get</i> .
remotehelp	Same as <i>help</i> , but the remote server rather than the local server is used to obtain help.
rename <i>old new</i>	Rename the file specified by <i>old</i> from <i>old</i> to <i>new</i> .
reset	Clear the reply queue.

continues &gt;&gt;

&gt;&gt;continued

Command	Description
<code>rmdir <i>dir</i></code>	Remove the directory on the remote server specified by <i>dir</i> .
<code>runique</code>	Toggle unique filenames. If it is on, and a duplicate filename is found, the new file will have a <i>.n</i> appended to it, where <i>n</i> is an integer that increments up starting at 1.
<code>send <i>file</i></code>	Same as <code>put</code> .
<code>status</code>	Display current status.
<code>sunique</code>	Remote unique file naming. Refer to <code>runique</code> , earlier in this table.
<code>type <i>type</i></code>	Set transfer type to <i>type</i> . Either binary (image) or ASCII (text).
<code>user <i>username</i></code>	Give username to the remote system when logging in.
<code>verbose</code>	Toggle verbose mode. If verbose mode is on, all FTP server responses are shown.

### FTP Autologin

A `.netrc` file can be placed in a user home directory to allow ftp file transfers to be automated to some extent. The file contains information about the connection including the hostname, login name, password, and other optional information.

The general format is as follows:

```
machine hostname login loginname password password
```

Consider the following example:

```
machine ftp.remotehost.com login anonymous password
➔user@localhost.com
```

It is very important that `.netrc` file permissions are set so that other users cannot access the file because it contains account names and passwords. Use the `chmod` command to set the `.netrc` file to mode 600.

►See Also `rcp` (90)

## rcp

```
rcp /usr/bin/rcp [options] [files]
```

---

Copy files across a network. In order to use `rcp`, remote commands must be allowed by using `rsh`. This requires the use of a `.rhosts` file or `/etc/hosts.equiv`. See `rsh` for details.

---

Options	Description
-p	Attempt to preserve all the file properties of the original, giving the copy the same time, mode, and ACL if possible.
-r	Recursively copy the directory specified by the <i>files</i> argument.

### Arguments

The arguments of rcp are specified as remote files or local files.

For local files:

path

For remote files:

host:path

user@host:path

### rcp Warning

rcp cannot properly copy directories containing symbolic links. A possible alternative is to use `cpio` to pipe the directory to `rcp`.

### Example

```
rcp remotehost:testfile testfile
```

### Description

Copy a file from a remote machine to the local host.

```
rcp remote1:report.txt
➔remote2:newreport.txt
```

Copy a file between two remote machines (third-party copy).

►See Also `rsh` (99), `ftp` (87)

## rwho

```
/usr/bin/rwho [options]
```

rwho

---

List who is logged on to machines on the local network. Output is similar to that of `who`. The remote server must be running the `in.rwhod` daemon.

---

►See Also `finger` (79), `rusers` (84), `rwho` (91)

## Communications

### mail

mail /usr/bin/mail [*options*] [*recipient*]

Utility to read or send (using `sendmail`) electronic mail to users on the local system and over the Internet. If mail is run with no recipient given, it can be used to read and process mail. A question mark (?) will be given as a command prompt. If mail exits while a message is being composed, it will be saved to `dead.letter` in the current directory. Incoming mail is saved in `/var/mail/username` or at a location specified in the `$MAIL` variable, where `username` is the current login name of the user running mail.

Option	Description
-e	Mail is not printed. However, mail returns an error code: <ul style="list-style-type: none"> <li>■ 1 if there is no mail.</li> <li>■ 0 if there is mail.</li> </ul>
-f <i>file</i>	Use the file specified by <i>file</i> as the mailfile rather than the default.
-h	Show headers instead of latest message when starting.
-m <i>type</i>	Add a header to the message of the form: <code>Message-Type: <i>type</i></code>
-p	Print all messages without checking dispositions.
-P	Print all headers when displaying messages.
-q	Quit and exit when an interrupt is received.
-r	Print messages in “first-in, first-out” order.
-t	Add a header to the message (for each recipient) of the form: <code>To: <i>recipient</i></code>
-w	Do not wait for remote transfer program to exit before sending.
-x <i>level</i>	Set debugging level to <i>level</i> (creates a tracefile in <code>/tmp</code> ).

Command	Description
?	Help. Displays all commands with their usages.
#	Print the current message number.
-	Previous message.
+ or newline	Next message.
!command	Use the shell to run command.
a	Display a message that arrived since mail was started.
d	<ul style="list-style-type: none"> <li>■ d or dp: Delete the current message and display the next one.</li> <li>■ d <i>n</i>: Delete message number <i>n</i>.</li> <li>■ dq: Delete the current message and then quit.</li> </ul>
h	<ul style="list-style-type: none"> <li>■ h: Show headers for current message.</li> <li>■ h <i>n</i>: Show headers for message number <i>n</i>.</li> <li>■ h a: Show headers for all messages.</li> <li>■ h d: Show headers for messages marked for deletion.</li> </ul>
m <i>recipient</i>	Send the current message to <i>recipient</i> , and delete it.
number	Go to message number <i>n</i> .
p	Reprint the current message, ignoring non-printable characters.
P	Reprint current message showing all headers.
q or CTRL+D	Quit. Any messages that were marked for deletion are not deleted.
r <i>recipients</i>	Reply to sender. If <i>recipients</i> are specified, they are carbon-copied on the mail.
s <i>mailfile</i>	Save messages in the file specified by <i>mailfile</i> . Default file is <code>mbox</code> .
u <i>n</i>	Undelete message number <i>n</i> . If no number is given, the last message read is undeleted.
w <i>file</i>	Write the current message to <i>file</i> , suppressing any headers. If no file is given, <code>mbox</code> is used.
x	Save all messages and exit.
y <i>file</i>	Same as w.

► See Also write (98)

## mail (UCB version)

```
mail /usr/ucb/mail [options] users
(UCB version) /usr/ucb/mail
```

---

Use the UCB version of the mail utility to read or send (using `sendmail`) electronic mail to users on the local system and over the Internet. If mail is run with no recipient given, it can be used to read and process mail. A `?` will be given as a command prompt. If mail exits while a message is being composed, it will be saved to `dead.letter` in the current directory. Incoming mail is saved in `/var/mail/username` or at a location specified in the `$MAIL` variable in which `username` is the current login name of the user running mail.

---

Option	Description
-B	No buffer. Neither standard input nor standard output.
-b <i>bcc</i>	Blind carbon copy. Blind carbon copy the users in the list <i>bcc</i> . More than one recipient may be listed, but they must be enclosed in quotes.
-c <i>cc</i>	Carbon copy. Carbon copy the users in the list <i>cc</i> . More than one recipient may be listed, but they must be enclosed in quotes.
-d	Debug. Extra debugging information is given.
-e	Test for mail. No output is given. A return value is given indicating the presence of mail: <ul style="list-style-type: none"> <li>■ 0: Mail</li> <li>■ 1: No mail</li> </ul>
-F	The message is recorded in a file with the same name as the recipient.
-f <i>msgfile</i>	Read messages out of the file specified by <i>msgfile</i> rather than the default mailbox file.
-H	Header summary. Only message headers are shown.
-h <i>num</i>	Set the maximum number of “network hops” to <i>num</i> . Setting this option will prevent endless mail loops.
-I	Include newsgroup and article-id headers.
-N	No initial header summary.
-n	The system default rc files ( <code>mailx.rc</code> and <code>Mail.rc</code> ) are not read upon startup.
-r <i>addr</i>	Set the message return address to <i>addr</i> .

*continues >>*

&gt;&gt;continued

Option	Description
-s <i>subject</i>	Set the message subject to <i>subject</i> . To be safe, the subject should be enclosed in quotes.
-T <i>file</i>	The message-id and article-id headers are saved to the file specified by <i>file</i> .
-t	Obtain To:, cc:, and bcc: fields from the message text rather than from the command line recipient list.
-u <i>user</i>	Use the mailbox of <i>user</i> rather than the current user mailbox.
-V	Display mail version number and exit.

Command	Description
!cmd	Execute the given shell command. If no command is given, a command shell is spawned—type <b>exit</b> to return to mail.
=	Display the current message number.
?	Display command summary.
alias <i>alias</i> <i>recip</i>	Make an alias for the specified mail recipient. Usually aliases are defined in the <code>.mailrc</code> file.
cd <i>dir</i>	Change directory to <i>dir</i> .
copy <i>file</i>	Same as the <code>save</code> command. However, the message is not marked as saved.
delete <i>msgs</i>	Delete the listed messages. If no messages are listed, the current message is deleted.
discard ↳ <i>headers</i>	Discard specified <code>headers</code> when displaying messages.
ignore <i>headers</i>	Do not print the specified <code>headers</code> when displaying messages.
dp <i>msgs</i>	Delete-print. Delete the specified messages and then print the next message.
edit <i>msgs</i>	Edit the messages using the editor specified in the <code>EDITOR</code> environment variable.
exit	Exit mail.
field <i>header</i>	Display the contents of the header specified by <i>header</i> .
file <i>file</i>	Switch mailbox files to <i>file</i> . If no file is specified, the current filename is displayed.

continues &gt;&gt;

&gt;&gt;continued

Command	Description
<code>followup msg</code>	Reply to the author of the specified message.
<code>from msgs</code>	Print the header summary of <i>msgs</i> . If no messages are specified, print the header summary for the current message.
<code>hold msgs</code>	Hold the listed messages in the current mailbox.
<code>inc</code>	Incorporate new messages that arrive in the current mail list.
<code>load file</code>	Load the specified file as a mail message. The file must be in a standard single message format, with headers.
<code>mail recip</code>	Send a message to the specified recipient.
<code>Mail recip</code>	Send a message to the specified recipient and save a copy.
<code>more msgs</code>	Display the listed messages, pausing after each page.
<code>Unread msgs</code>	Mark the listed messages as unread.
<code>next</code>	Jump to the next message in the message list.
<code>pipe msgs</code> <code>command</code>	Pipe the listed messages through the specified command.
<code>print msgs</code>	Print the specified messages.
<code>put msgs file</code>	Write the specified messages to the specified file.
<code>quit</code>	Quit. Messages that were saved in a file are deleted; all others are saved.
<code>reply msgs</code>	Reply to each message specified. The subject line is taken from the first message in the list of messages.
<code>replyall msg</code>	Reply to the message, sending a copy to every recipient of the original message.
<code>Save</code>	Save the specified message, all addressing information is stripped.
<code>save</code>	Save the specified message to the mailbox file (usually <code>mbox</code> ).
<code>source file</code>	Execute the commands in the specified file.
<code>shell</code>	Spawn a shell to run commands. Type <code>exit</code> to return to mail.
<code>top msgs</code>	Display only the first few lines ( <code>top</code> ) of the listed messages.
<code>unalias</code> <code>→aliases</code>	Remove an alias definition. Refer to <code>alias</code> , earlier in this table.
<code>undelete msgs</code>	Undelete the listed messages.
<code>unignore</code>	No longer ignore the specified headers. Refer to <code>ignore</code> , earlier in this table.
<code>version</code>	Display the current version of mail.
<code>visual</code>	Edit the message in visual mode. The editor given in the <code>VISUAL</code> environment variable is used. If this variable is not set, <code>vi</code> is used.

## rwall

`/usr/sbin/rwall [options] [host]`

rwall

Broadcast a message to all users on a network. The message, when received, will be preceded by the following header:

```
Broadcast Message...
```

The remote server must be running the `walld` daemon.

Option	Description
<code>-n netgroup</code>	Broadcast the message to the network specified by <i>netgroup</i> rather than to a specific host.
<code>-h host</code>	Specify a single host to broadcast the message to. This option can be used in conjunction with the <code>-n</code> option.

►See Also `rwho` (91), `rusers` (84)

## talk

`/usr/bin/talk user [terminal]`

talk

Talk to other users using the UNIX talk protocol. It can be used to chat with another user on the same system or with other users across the Internet. The remote system must be running a talk daemon (`in.talkd`) listening on port 517/udp.

When connecting to a remote system, the other user will be prompted to talk with a message similar to the following:

```
Message from Talk_Daemon@remotehost at 8:55 ...
talk: connection requested by
mulligan@localhost.
```

```
talk: respond with: talk mulligan@localhost
```

After the other user responds, chatting can begin. The text is sent character by character.

Argument	Description
<i>user</i>	The login name of the user to talk to. It can either be a username (for the local system), or an Internet address of the form <i>user@remotehost</i> for Internet chatting.
<i>terminal</i>	Optional. If a user is logged in more than once, this specifies which terminal to talk to.

Command	Description
<b>CTRL+L</b>	Refresh the screen.
<b>CTRL+D</b> or <b>CTRL+C</b>	End the talk session and quit.

►See Also mail (92), write (98)

## write

write /usr/bin/write [*terminal*]

---

Send a message to another user on the local system. Unlike `talk`, the text is sent line by line.

When used, `write` will send a message similar to the following to the other user:

```
Message from mulligan on host (pts/6) [ Tue Mar  4 09:11:22 ]
➤...
```

To stop sending messages, press **CTRL+D** or **CTRL+C**.

---

Argument	Description
<i>terminal</i>	Optional. If the other user is logged in more than once, this specifies which terminal to send the message to.

►See Also mail (92), talk (97)

## Remote Shells and Login

### rlogin

rlogin /usr/bin/rlogin [*options*] [*host*]

#### Description

---

Log in to a remote system using the remote login procedures rather than telnet. The user and/or host must be defined in a `.rhosts` or `hosts.equiv` file on the remote system. The remote server must be running the `in.rlogind` daemon. This is an unencrypted and relatively insecure method of remote login.

---

Option	Description
-L	Litout mode.
-8	Use 8-bit data instead of 7-bit.
-e <i>char</i>	Set escape character to <i>char</i> .
-l <i>username</i>	Use <i>username</i> as the username for remote login rather than the current one.

►See Also `rcp` (90), `telnet` (100)

## rsh

`/usr/bin/rsh [options] [hostname] [command]` rsh

The remote shell executes commands on a remote system across the network. Alternatively, if no command is given, `rsh` behaves like `rlogin` and will log the user in to the remote system. This command uses the `.rhosts` or `/etc/hosts.equiv` file on the remote system to check if remote execution/login is allowed.

Option	Description
-l <i>username</i>	Use the username specified by <i>username</i> for logging in rather than the current username.
-n	Suppress standard output from <code>rsh</code> . Output is sent to <code>/dev/null</code> .

### .rhosts and hosts.equiv

Two files are used to check authentication for the `rlogin`, `rsh`, and `rcp` programs: `.rhosts` and `hosts.equiv`. The `.rhosts` file is placed in a user home directory and lists the hosts and users that are allowed to log into the local account. The general form of the `.rhosts` file is as follows:

```
hostname username
```

If the username is omitted, then all users from the specified host are allowed. Optionally, a *netgroup* can be used in place of the username by specifying it as `+@netgroup`. The symbol `+` is used to represent “all.” For example, the following allows remote logins from the user `smithj` from any host:

```
+ smithj
```

However, this leaves the account open to security exploits and should not be used.

*continues >>*

*continued >>*

The `/etc/hosts.equiv` file can be thought of as an `.rhosts` file for the entire system. If a remote user is allowed to log in based on the `host.equiv` file, she will be allowed to log in as any local user.

Note that the `.rhosts` and `/etc/hosts.equiv` files are two of the most common sources of security breaches. To be safe, follow these recommendations:

- Always use the `hostname username` form; never specify only a user or a host.
- Never use a `+` in an entry.
- Only list hosts that are completely trusted and very secure.
- Check `.rhost` and `host.equiv` files on a regular basis for signs that they have been altered or tampered with in any way.

### Example

Display a text file that is located on a remote system:

```
rsh remotehost cat file.txt
```

►See Also `rlogin` (98)

## telnet

```
telnet /usr/bin/telnet [options] [host] [port]
```

---

Interface to the `telnet` protocol to log in to a remote system. `telnet` can also be used to connect to an arbitrary port/service on a remote host.

---

Option	Description
<code>-8</code>	Use 8-bit data path.
<code>-E</code>	Suppress all escape characters.
<code>-L</code>	Use 8-bit data path on output.
<code>-c</code>	Do not read <code>.telnetrc</code> file.
<code>-d</code>	Toggle debugging mode on.
<code>-e char</code>	Set escape character to <code>char</code> .
<code>-l user</code>	Send the current username as the value of <code>user</code> .
<code>-n file</code>	Open <code>file</code> as a tracefile for the session.
<code>-r</code>	Force <code>telnet</code> to behave more like <code>rlogin</code> . Escape characters: <code>~</code> , <code>.Ar</code> , and <b>CTRL+Z</b> can be used (see <code>rlogin</code> ).

Command	Description
<b>Ctrl-]</b>	Drop back to a prompt. This is very useful when a connection is hung and needs to be terminated. Simply press <b>Ctrl+]</b> and then type <b>quit</b> .
? <i>command</i>	Help. Get help on <i>command</i> . If <i>command</i> is not specified, then a list of commands will be given.
close	Close the current session and exit.
display <i>arg</i>	Show values of parameters set by toggle.
environ <i>args</i>	Set variables that can be sent to the remote host through environment variables. Arguments include the following: <ul style="list-style-type: none"> <li>■ define <i>variable value</i></li> <li>■ undefine <i>variable</i></li> <li>■ export</li> <li>■ unexport <i>variable</i></li> <li>■ list</li> <li>■ ?</li> </ul>
logout	Same as close, if logout is supported on the remote side.
open <i>host port</i>	Open connection to host on port. If no port is given, 23/tcp is used. Optionally, -l user can be added to specify a username other than the current one.
quit	Same as close.
send <i>args</i>	Send <i>args</i> (including special characters) to the remote host. Arguments include the following: escape, synch, brk, ip, abort, ao, ayt, ec, el, eof, eor, ga, getstatus, nop, susp.
set or unset <i>arg value</i>	Set or unset any of the following arguments: <ul style="list-style-type: none"> <li>■ echo: Toggle local echoing.</li> <li>■ escape: Set escape character to value.</li> <li>■ interrupt: Set interrupt character to <i>value</i>.</li> <li>■ quit: Set quit character.</li> <li>■ flushoutput: Set flushoutput character.</li> <li>■ erase: Set erase character.</li> <li>■ eof: Set eof character.</li> </ul>

continues >>

&gt;&gt;continued

Command	Description
	<ul style="list-style-type: none"> <li>■ ayt: Set the Are You There character.</li> <li>■ lnext: Set lnext character in the old line-by-line mode.</li> <li>■ reprint: Set the reprint character.</li> <li>■ rlogin: Set rlogin escape character.</li> <li>■ start: Set start character.</li> <li>■ stop: Set stop character.</li> <li>■ susp: Set the suspend character.</li> <li>■ tracefile: Set the trace file.</li> <li>■ worderase: Set the worderase character.</li> </ul>
status	Display status of telnet.
toggle args	Toggles on/off the following parameters: autoflush, autosynch, binary, inbinary, outbinary, crlf, crmod, debug, localchars, netdata, options, prettydump, skiprc, termdata Toggling ? will show all available parameters.
z	Suspend telnet. Job control must be supported. ►See Also rlogin (98)