

Executing Commands on a Remote Host [3]

This section describes how you can use TCP/IP utilities to execute commands on other computers on your network. Table 1 lists the utilities and summarizes the introductory information about accessing hosts provided in subsection 2.2.1, page 12. Use this information to choose a utility that will allow you to complete your task, then turn to the appropriate subsection for that utility.

Table 1. Features of TCP/IP utilities for accessing remote hosts

Features	telnet(1B)	rlogin(1B)	rsh (see remsh(1B))
Passes terminal type to remote host	Yes	Yes	Yes
Automatic login	No [†]	Yes	Yes
Establishes a virtual connection	Yes	Yes	No (only one command is executed on the remote host)

[†] Yes, if Kerberos is used in conjunction with `telnet`.

Using the telnet utility

3.1

You can use the `telnet` utility in one of the following modes:

- Input mode
- Command mode

When the utility is in input mode, every command you type is executed as though your terminal were directly connected to the remote host. While in command mode, the local `telnet` utility interprets special `telnet` commands that are typed after a `telnet` prompt (`telnet>`). Read the following subsections to learn how to use both modes of `telnet`.

Using telnet in input mode

3.1.1

The format of the telnet(1B) utility is as follows:

```
telnet [-a] [-d] [-n tracefile] [host [port]]
```

<i>-a</i>	Tries automatic login by using Kerberos.
<i>-d</i>	Sets the initial value of the debug toggle to TRUE. (For more information on the debug command, see page 27.)
<i>-n tracefile</i>	Opens the specified file for recording trace information. (For more information on the set tracefile command, see page 26.)
<i>host</i>	Indicates the official name, an alias, or the Internet address of a remote host.
<i>port</i>	Indicates a port number (address of an application). If you omit a number, the default telnet port is used.

When in input mode, you can log in to and use the resources of a remote host. The two forms of input mode are character-at-a-time or line-by-line; the mode used depends on which mode the remote host supports. In character-at-a-time mode, most typed text is sent immediately to the remote host for processing. In line-by-line mode, all text is echoed locally, but only completed lines of text are sent to the remote host. You can use the local echo character (initially `CONTROL-e`) to turn off and on the local echo. Most often, this would be used to enter passwords without the password being echoed, and needed only when the remote side does not properly support line-by-line mode. If you connect to a machine that does not support a line-mode telnet server, do an explicit mode line command to force you into kludge line mode. You then can use `CONTROL-e` to turn local echo on and off. Press `CONTROL-e` `CONTROL-d` to enter kludge line mode immediately.

The steps for entering input mode are as follows:

1. Type the telnet command and the name of the remote host to which you want to connect. You can use either the official host name, one of its aliases, or the Internet address to identify the remote host. Press `RETURN`. In the following example, the remote host is `remote_host`.

```
$ telnet remote_host
Trying...
Connected to remote_host.
Escape character is '^]'.
remote operating system (remote_host)
login:
```

As shown, after you execute the `telnet` command, the utility displays text as it tries to make the connection. Line 3 shows that the connection is made. Line 4 displays the default escape character, `^]` (or `[CONTROL-]`), which is used to enter command mode. Line 5 shows the remote host's operating system and the name of the remote host. Line 6 is the login prompt.

2. Type your login name for the remote host and press `[RETURN]`, as follows:

```
login: my_name
password:
```

3. Type your remote-system password after the prompt and press `[RETURN]`. Your password does not appear on the screen. This is a security measure.

The remote host checks the system authorization files before granting you access. You must have a login account on the remote host to access it through `telnet`. Consult with your system administrator for help in setting up accounts and user profiles on remote hosts.

4. When you complete the login process, a message is issued and the remote host's prompt appears on the screen, as follows:

```
Last successful login was: Tue Aug 23 14:00:31
from Cray Research host
%
```

Now you have a virtual connection and can use the resources of the remote host as though your terminal were directly attached to it.

Usually, the remote host determines whether the connection will operate in character-at-a-time or line-by-line mode. You can use the `telnet status` command to determine which mode is currently being used. You can manually override the mode by using the `telnet mode` command. For more information on the use of these commands, see subsection 3.1.3, page 23.

If you are using a slow network to get to the remote host, it is useful to switch from character-at-a-time to line-by-line mode if the remote host supports it. This switch causes no loss of functionality and eliminates the generation of network traffic for each character typed, resulting in better command response time. Even if the remote host does not support line-by-line mode, you can still use it. In this case, you must manually disable character echoing on the remote system; consequently, you cannot use any visually oriented commands (such as a full-screen editor) while in line-by-line mode.

After you complete your work on the remote host, terminate input mode by using the remote host's normal logout procedure. When you log out from the remote host, the `telnet` utility returns you to your working directory on the Cray Research system, with the following message:

```
Connection closed by foreign host.  
$
```

Using telnet in command mode

3.1.2

The `telnet` command mode is not used to execute commands on a remote host, but rather, to execute `telnet` commands for the purpose of opening or closing a connection, displaying `telnet` information, or changing the conditions of the `telnet` utility. You can enter the `telnet` command mode in the following ways:

- If you are in your working directory on the Cray Research system, type `telnet` without a remote host name.
- If you are in `telnet` input mode, press the escape character.

Use the following procedure to enter `telnet` command mode from your working directory on a Cray Research system:

Type `telnet` and press `RETURN`. The utility responds with the `telnet` prompt to indicate that you are in command mode and that the `telnet` utility is ready to execute a `telnet` command.

```
$ telnet
telnet>
```

Enter a `telnet` command (see subsection 3.1.3, page 23, for a list of available commands), using the following format:

```
telnet> command
```

For example, to display the status of the `telnet` utility, type the `status` command, as follows:

```
telnet> status
No connection.
Operating in line-by-line mode.
Escape character is '^]'
telnet>
```

Use the following procedure to enter command mode from input mode:

Enter the escape character `CONTROL-]`, as follows:

```
%^]
telnet>
```

Usually, a connection is closed by logging out from the remote system. If for some reason you cannot do that, use the `close` command to close the `telnet` connection, as follows:

```
telnet> close
Connection is closed.
telnet>
```

To exit the `telnet` utility completely, type the `quit` command and press `RETURN`. The Cray Research system prompt then appears, as follows:

```
telnet> quit
$
```

Use the following procedure to enter input mode from command mode:

Type the `open` command and the name of the remote host. Press `RETURN`.

```
telnet> open my_host
Trying...
Connected to my_host
Escape character is '^]'.
remote operating system (my_host)
login:
```

Type your login name and press `RETURN`. Type your password and press `RETURN`. The `telnet` utility then establishes a connection, and the remote host displays its system prompt. Now you can work on the remote host.

Note: Invoking the `open` command from command mode is the same as entering `telnet` input mode. Both establish a virtual connection and let you use the remote host.

telnet commands

3.1.3

This subsection lists the `telnet` commands and describes their functions. You can abbreviate command names to the minimum number of characters required to uniquely identify the command. In the following list, brackets enclose the part of a command name that you can omit.

- `c[lose]` Closes a `telnet` session and returns to command mode.
- `d[isplay] arguments` Displays all, or some, of the set and toggle values (see values for set and toggle commands).
- `m[ode] type` The *type* argument is either `line` (for line-by-line mode) or `character` (for character-at-a-time mode). Requests permission from the remote host to go into the requested mode. If the remote host can enter that mode, the requested mode is entered.
- `o[pen] host [port]` Opens a connection to the specified host. If you do not specify a port number, `telnet` tries to contact a `telnet` server at the default port. The host specification can be either a host name (see `host(1B)`) or an Internet address specified in the “dot notation” (see `inet(3)`).
- `q[uit]` Closes any open `telnet` session and exits `telnet`. An end-of-file (EOF) character (in command mode) also closes a session and exits.
- `sen[d] arguments` Sends one or more special character sequences to the remote host. The following are the arguments that can be specified (more than one argument can be specified at a time):

<u>Argument</u>	<u>Description</u>
<code>ao</code>	Sends the <code>telnet</code> AO (Abort Output) sequence, which causes the remote system to flush all output from the remote system to the user's terminal.
<code>a[yt]</code>	Sends the <code>telnet</code> AYT (Are You There) sequence, to which the remote system may choose to respond.
<code>brk</code>	Sends the <code>telnet</code> BRK (Break) sequence, which might have significance to the remote system.

<u>Argument</u>	<u>Description</u>
ec	Sends the telnet EC (Erase Character) sequence, which causes the remote system to erase the last character entered.
el	Sends the telnet EL (Erase Line) sequence, which causes the remote system to erase the line currently being entered.
es[cape]	Sends the current telnet escape character (initially ~).
g[a]	Sends the telnet GA (Go Ahead) sequence, which probably has no significance to the remote system.
ip	Sends the telnet IP (Interrupt Process) sequence, which cause the remote system to abort the currently running process.
n[op]	Sends the telnet NOP (No OPeration) sequence.
s[ynch]	Sends the telnet SYNCH sequence, which causes the remote system to discard all previously typed (but not yet read) input. This sequence is sent as TCP urgent data (and might not work if the remote system is a 4.3BSD system). If the SYNCH sequence does not work, a lowercase r might be echoed on the terminal.
?	Prints out help information for the send command.

set variable values

Sets any one of a number of `telnet` variables to a specific value. The special value `off` turns off the function associated with the variable. The values of variables can be interrogated with the `display` command. The variables that you can specify are as follows:

<u>Variable</u>	<u>Description</u>
<code>ec[ho]</code>	This is the value (initially <code>CONTROL-e</code>) which, when in line-by-line mode, toggles between doing local echoing of entered characters for standard processing, and suppressing echoing of entered characters (for example, for entering a password).
<code>eo[f]</code>	If <code>telnet</code> is operating in line-by-line mode, entering this character as the first character on a line causes this character to be sent to the remote system. The initial value of the EOF character is taken to be the terminal's EOF character.
<code>er[ase]</code>	If <code>telnet</code> is in <code>localchars</code> mode (see <code>toggle localchars</code> that follows), and if <code>telnet</code> is operating in character-at-a-time mode, a <code>telnet EC</code> sequence (see <code>send ec</code> preceding) is sent to the remote system when this character is typed. The initial value for the erase character is taken to be the terminal's erase character.
<code>es[cape]</code>	This is the <code>telnet</code> escape character (initially <code>CONTROL-]</code>), which causes entry into <code>telnet</code> command mode (when connected to a remote system).
<code>i[nterrupt]</code>	If <code>telnet</code> is in <code>localchars</code> mode (see <code>toggle localchars</code> that follows), and the interrupt character is typed, a <code>telnet IP</code> sequence (see <code>send ip</code> , under <code>send arguments</code> , preceding) is sent to the remote host. The initial value for the interrupt character is taken to be the terminal's interrupt character.

<u>Variable</u>	<u>Description</u>
k[ill]	If telnet is in localchars mode (see toggle localchars that follows), and if telnet is operating in character-at-a-time mode when this character is typed, a telnet EL sequence (see send el, under send arguments) is sent to the remote system. The initial value for the kill character is taken to be the terminal's kill character.
q[uit]	If telnet is in localchars mode (see toggle localchars that follows), and the quit character is typed, a telnet BRK sequence (see send brk preceding) is sent to the remote host. The initial value for the quit character is taken to be the terminal's quit character.
t[racefile]	<i>tracefile</i> If either netdata or options was toggled to be TRUE, the debugging information generated by them is written into <i>tracefile</i> . If <i>tracefile</i> is given as -, the debugging information is written to standard output (the default case).
st[atus]	Shows the current status of telnet. The information displayed includes the name of the peer to which you are connected, as well as the current mode.
t[oggle] arguments	Toggles (between TRUE and FALSE) various flags that control the manner in which telnet responds to events. You can specify more than one argument. To interrogate the state of these flags, use the display command. Valid arguments are as follows:
<u>Arguments</u>	<u>Description</u>
autof[lush]	If autoflush and localchars are both TRUE, when the interrupt or quit characters are recognized (and transformed into telnet sequences; see the entry for set variable values for details), telnet refuses to display any data on the user's terminal until the remote system acknowledges (by way of a telnet Timing Mark option) that it has processed those telnet sequences. The initial value for this toggle is TRUE (see stty(1)).

<u>Arguments</u>	<u>Description</u>
<code>autos[ynch]</code>	If <code>autosynch</code> and <code>localchars</code> are both TRUE, when either the interrupt or quit characters are typed (see the entry for <code>set variable values</code> for descriptions of the interrupt and quit characters), the resulting telnet sequence sent is followed by the telnet SYNCH sequence. This procedure should cause the remote system to begin discarding all previously typed input until both telnet sequences have been read and acted upon. The initial value of this toggle is FALSE.
<code>crm[od]</code>	Toggles carriage return mode. When this mode is enabled, most carriage return characters received from the remote host are to be mapped into a carriage return followed by a line feed. This mode does not affect those characters that the user types; it affects only those received from the remote host. This mode is not very useful unless the remote host sends carriage returns, but never line feeds. The initial value for this toggle is FALSE.
<code>d[ebug]</code>	Toggles socket-level debugging (useful only to the super user). The initial value for this toggle is FALSE.
<code>l[ocalchars]</code>	If this is TRUE, the interrupt, quit, erase, and kill characters (see <code>set variable values</code> , preceding) are recognized locally, and transformed into appropriate telnet control sequences (respectively, BRK, EC, EL, and IP; see the entry for <code>send arguments</code> , preceding). The initial value for this toggle is TRUE in line-by-line mode, and FALSE in character-at-a-time mode.
<code>n[etdata]</code>	Toggles the display of all network data (in hexadecimal format). The initial value for this toggle is FALSE.

<u>Arguments</u>	<u>Description</u>
o[ptions]	Toggles the display of internal telnet protocol option processing. The initial value for this toggle is FALSE.
?	Displays the legal toggle commands.
z	Suspends the telnet command and executes a subshell.
? [command]	Gets help. Without arguments, telnet prints a help summary. If a command is specified, telnet prints the help information for the specified command.

Using the rlogin utility

3.2

The rlogin utility lets you automatically log in to a remote host on which you have an account. It establishes a virtual connection; that is, your terminal appears to be physically connected to the remote host.

If you are accessing a remote host that does not support rlogin, you can use the telnet utility as an alternative login service.

Note: When the UNICOS multilevel security (MLS) feature is enabled, the autologin capability of inbound rlogin is allowed only if the following requirements are met:

- The NETW_RCMD_COMPAT configuration parameter is disabled. Check with your system administrator
- The client host is named in the /etc/host.equiv file.
- Your remote and local user IDs are identical.
- Your user ID is specified in the .rhosts file of the server.
- The client host is also specified in the .rhosts file of the server.
- The workstation access list (WAL) specifies login permission for one or more of the following: the remote host you want to access, your account name, or your group ID entry. See subsection 8.2.4, page 104, for more information.

The `rlogin` utility has the following format:

```
rlogin rhost [-ec] [-8] [-1 username]
```

<i>rhost</i>	Indicates the official name, an alias, or the Internet address of a remote host.
- <i>ec</i>	Lets you change the <code>rlogin</code> escape character from the tilde (~) to the new escape character (<i>c</i>). Do not type a space between the option character and the argument specifying the new escape character.
-8	Allows transmission of 8-bit data.
-1 <i>username</i>	Lets you log in automatically, even if your login name on the host to which you want access does not match your login name on the Cray Research system. To use this option, enter your remote-host login name after -1. Check with your system administrator about whether this option is disabled if you are on a UNICOS MLS system.

To log in to and out of another host, complete the following steps:

1. Type the `rlogin` command and the name of the remote host. You can specify the host's official name, an alias, or the Internet address. Press `RETURN`.

```
$ rlogin my_host  
Password:
```

2. The password prompt does not appear if you are authorized for automatic login. If the password prompt appears, type your remote-host password after the prompt. As a security measure, your password is not displayed on the screen. After `rlogin` establishes a connection, it displays system information and the remote host's prompt, as follows:

```
Password:  
Last successful login was:  
Wed Sep 7 13:16:14 from Cray Research host  
%
```

Now you can use the resources of the remote host.

3. To terminate the remote connection, type the remote host's logout command. (In the following example, the command is `exit`.) The `rlogin` utility then returns you to your working directory on the Cray Research system.

```
% exit  
Logout: Thu Sep 8 13:44:18 CDT 1988  
Connection closed.  
$
```

Other options that you can use with the `rlogin` command are explained in the following examples.

Example 1:

In this example, the local host is called `cray`, the remote host is `engineering`, and the user's login name on both hosts is `bonnie`. The user types the `rlogin` command followed by the name of the remote host and presses `RETURN`.

```
$ rlogin engineering  
Last login was: Fri Sep 9 15:56:53 from cray  
%
```

The remote host `engineering` checks its `/etc/hosts.equiv` file to determine whether host `cray` is listed. When it finds `cray`, it checks to see whether the login name `bonnie` is in its password file. It is, so the user is logged in automatically. User `bonnie` is not asked to provide a password because she already authenticated her account by entering her password when she logged in to her local host. When autologin is complete, the remote host's prompt appears on the screen.

For more information on the autologin feature, see subsection 7.1, page 77. Also see the preceding note in this subsection about using autologin with UNICOS MLS enabled.

Example 2:

In this example, the user's login name on the local host is adam, and his login name on the remote host (math) is apj. No account exists on the remote host with a login name adam. Because the user's login names are different on the two hosts, he must specify the `-l` option with the login name apj. The `-l` option tells the `rlogin` utility to look for the user's local host and login name in the `.rhosts` file for apj in the home directory on math. When the entry is found, the user is automatically logged in to the remote host.

```
$ rlogin math -l apj
Last successful login was:
Fri Sep 9 14:51:58 from cray
%
```

Example 3:

In the following example, the user logs in to remote host chemistry and changes the escape character from `~` to `@`. To do this, the user types the `rlogin` command, the remote host name, and the `-e` option followed by the new escape character, `@`, and presses `RETURN`. As demonstrated, the utility recognizes `@` as the new escape character and closes the connection.

```
$ rlogin chemistry -e@
Last successful login was:
Fri Sep 9 14:51:58 from cray
% @.
Closed connection.
$
```

Using the rsh utility

3.3

The `rsh` (remote shell) utility initiates a login to a remote host and executes a command. Like `rlogin`, `rsh` gives you automatic authorization to access your accounts on remote hosts. When you use the `rsh` utility to execute a command on a remote host, interrupt, quit, and terminate signals are passed to the remote host. The remote login terminates when the command finishes processing; you are then returned to your working

directory on the Cray Research system. Because the `rsh` utility lets you invoke only a single shell script before returning you to the Cray Research system, you cannot use interactive commands such as `vi`. For interactive applications, use `telnet` or `rlogin`.

If automatic authorization is not available on your system, you can use either the `rlogin` or `telnet` utility as an alternative to `rsh`.

Note: When the UNICOS multilevel security (MLS) feature is enabled, the use of the `rsh` utility is allowed only if the following requirements are met:

- The `NETW_RCMD_COMPAT` configuration parameter is disabled. Check with your system administrator.
- The client host is named in the `/etc/host.equiv` file.
- Your remote and local user IDs are identical.
- Your user ID is specified in the `.rhosts` file of the server.
- The client host is also specified in the `.rhosts` file of the server.
- The workstation access list (WAL) specifies `rsh` permission for one or more of the following: the remote host you want to access, your account name, or your group ID entry.

The `rsh` utility, located in the `/usr/ucb` directory, has an identically named, but entirely different, `rsh` (restricted shell) command in the `/bin` directory. Therefore, `remsh`, an alternative name for the remote shell utility, is also located in `/usr/ucb`. Thus, to be certain you are using the remote shell utility, you can either use `remsh` rather than `rsh`, or place the `/usr/ucb` directory before the `/bin` directory in your search path (determined by the `$PATH` environment variable).

Note: Shell metacharacters (`&`, `?`, `*`, `|`, `\`, `;`, `<`, and `>`) that are entered without quotation marks around them are interpreted on the Cray Research system; quoted metacharacters are interpreted on the remote host.

Use the following format to invoke the `rsh` command:

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

- host* Indicates the official name, an alias, or the Internet address of a remote host.
- `-l username` Lets you log in automatically, even if your login name on the host to which you want access does not match your login name on the Cray Research system. To use this option, enter your remote-host login name after `-l`. See the note at the beginning of this subsection for requirements that must be met to use this option.
- `-n` If no input is desired, you must use `-n` to redirect the input of `rsh` to the `/dev/null` file.
- command* A UNIX command or the name of a script on the remote host. If you omit this argument, and the only argument to `rsh` is *host*, an `rlogin` command is executed to establish a connection to *host*.

These options and arguments are explained in the following examples.

Example 1:

In this example, the user types the `rsh` command, the remote host name `bio`, the `who` command, and presses `RETURN`. The `rsh` utility performs an autologin to remote host `bio` and executes the `who` command, displaying information on the user's terminal. After the command is executed, the utility returns the user to his or her working directory on the local host and displays the local system prompt.

```
$ rsh bio who
bonnie  ttyd1   Sep   9 15:34
bdh     ttyd7   Sep   9 15:07
deb     ttyd1   Sep   9 16:00   (rei-sc)
adam    ttyd3   Sep   9 11:15   (lanman)
jeni    ttyd3   Sep   9 16:22   (social-gate)
$
```

Example 2:

In this example, a user on local host `cray` wants to execute the `ls` command on remote host `chemistry` in order to see the files in his remote host directory. The user's login name on the Cray Research system is not the same as the login name of his account on the remote host.

The user types `rsh`, the remote host's name (`chemistry`), the `-l` option followed by the login name `scott`, the `ls` command, and presses `RETURN`. The `-l` option flags the `rsh` utility to check the `.rhosts` file in the home directory of user `scott`. The utility looks for the host name of the local host and the login name of the user initiating the remote login request. When they are found, the user is logged in automatically. The system executes the `ls` command, displays the file names on the screen, then returns the user to his working directory on the Cray Research system.

See the note at the beginning of this subsection for requirements to use the `-l` option.

```
$ rsh chemistry -l scott ls
data
memo
memo.sched
statistics
$
```