

Communicating Across the Network [5]

The following utilities are available for communicating across a TCP/IP network:

- `mail(1)` and `mailx(1)`

The `mail` and `mailx` utilities let you send and receive mail. When you send mail, `mailx` enables you to edit, review, and perform other modifications to the message as you enter it. When you read mail, both `mail` and `mailx` let you save, delete, and respond to messages.

- `talk(1B)`

The `talk` utility enables you to communicate with another user. This utility copies lines from your terminal to that of the other user.

The following subsections explain how to use these TCP/IP utilities.

Using the `mail` and `mailx` utilities

5.1

This subsection explains how to send mail messages and how to receive (or read) mail messages. The `mail` and `mailx` utilities are used for both sending and receiving.

Note: For more information about `mail` and `mailx` if you have a UNICOS MLS system or a Trusted UNICOS system, see subsection 8.3, page 112.

Note that you cannot trust mail “From” lines. The mail system is designed to deliver mail worldwide across a huge collection of systems that cannot be trusted. In most cases, you can count on the mail you receive to be from the person whose name is in the header, but this is not guaranteed. It is relatively simple to forge mail when it is sent across the network. You should always verify through an alternative communication method the source and context of any sensitive mail message.

Sending mail messages 5.1.1

To send mail to a user on your local host, use the following formats:

```
mail login_name
mailx login_name
```

login_name Valid login name on the local host

When sending mail to a user on the network, you can specify the network address by adding an at-sign (@) and the remote host name to *login_name*. You can specify more than one addressee for the same message by separating the network addresses with a space. Use the following format:

```
mail login_name[@rhost] ...
mailx login_name[@rhost] ...
```

login_name Valid login name on the remote host.

@rhost Official host name, alias, or Internet address of a remote host; if @ is not included, mail and mailx do not recognize the host name.

After you press `[RETURN]`, mailx prompts you for a message subject. Enter a line identifying the subject of the message, then go to a new line to begin the text of your message. The mail command does not prompt for message subject; simply begin the text of your message. To end the message, go to a new line and enter `[CONTROL-d]`.

Unlike the mail command, mailx has associated commands. You can enter mailx commands while you are creating the message by beginning a line with the tilde (~) escape character, followed by one command letter and optional arguments. See the mailx(1) man page for a summary of these commands. Two of the most useful tilde escape commands, which you can use in mailx input mode, are ~r and ~v. These commands are described in the following list.

~r *filename* Reads the file called *filename* into the body of the message.

~v Lets you edit the current message by using a text editor; the default text editor is vi(1).

Example:

In this example, the user addresses mail to the user with the login name `pattie` on the remote host `cray2`, using the network mail address `pattie@cray2`. The system prompts for a subject on the following line. The user enters the subject, then begins the body of the message on the next line. At the end of the message, the user goes to a new line and presses `CONTROL-d`.

The EOT (end-of-transmission) message appears on the screen, followed by the shell prompt.

```
$ mailx pattie@cray2
Subject: Sending mail
I'm practicing sending mail, so tell me if you get this.
CONTROL-d
EOT
$
```

Reading mail messages

5.1.2

To read incoming mail, enter either `mail` or `mailx` without any arguments, as shown in the following formats:

```
mail
mailx
```

The `mail` and `mailx` commands let you change the way incoming mail is displayed and stored. They are described on the `mail(1)` and `mailx(1)` man pages.

Using the talk utility

5.2

The `talk` utility is a visual communication program that copies lines from your terminal to that of another user.

Note: The `talk` command is not supported on a Trusted UNICOS system. The `talk` command allows communication only between users who are at the same security label in a UNICOS MLS system. See the *UNICOS Multilevel Security (MLS) Feature User's Guide*, publication SG-2111, for more information.

This command has the following format:

```
talk address [terminal]
```

address User to whom you are sending the message

terminal Terminal name to be used if the user is
logged in on more than one terminal

The message recipient is notified that you are sending a message and is instructed to return a response that opens the connection. You then type your message. You and the recipient can communicate simultaneously if desired. To exit the `talk` facility, type an interrupt character. The cursor then moves to the bottom of the screen, and the command restores the terminal.

Example:

In this example, your network mail address is `curtis@host4`. You are going to send a message to the user at network mail address `jerry@host4`.

```
$ talk jerry@host4
```

The recipient receives the following message:

```
Message from TalkDaemon@host4 at 13:06...  
talk: connection requested by curtis@host4  
talk: respond with: talk curtis@host4
```

You receive the message:

```
Connection established
```

You can then begin typing your message:

```
Shall we get together for lunch today?
```

The recipient can type a reply, and the communication proceeds until you or the recipient types an interrupt character.