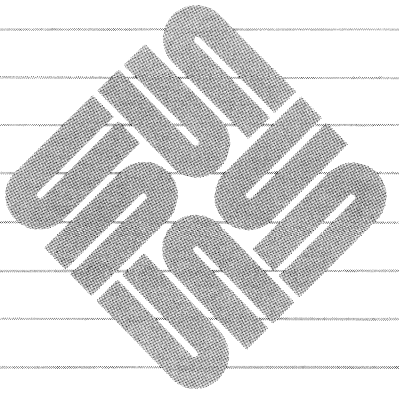
sun®
microsystems

Mail *and* Messages: Beginner's Guide



Part Number: 800-1709-10
Revision A, of 9 May 1988

Contents

Figures

Preface

This document introduces the electronic mail and message facilities available on the Sun workstation. We assume that you have some experience with the Sun Workstation, and the SunOS operating system.

We provide examples to learn how to send, read, and reply to mail and messages, not detailed explanations of the inner workings of the mail and message programs. However, as in each of the *Beginner's Guides*, we refer to the other Sun documentation, drawing a road map for you to follow when you wish to learn more about a certain topic.

Mail and Messages: Beginner's Guide describes how to send, read, store, reply to, and forward mail using the `mail` program. It introduces Mail Tool, a window and mouse mail program. In addition, this manual presents the various message programs and describes how mail travels over various networks. A command summary and a glossary provide easy access to the material.

Prerequisite documents

Getting Started with SunOS: Beginner's Guide

If you are running the SunView windows system or are planning to use Mail Tool (described here):

SunView 1 Beginner's Guide

Companion documents

Setting Up Your SunOS Environment: Beginner's Guide
Self-Help with Problems: Beginner's Guide
Doing More with SunOS: Beginner's Guide
Using the Network: Beginner's Guide

SunOS Reference Manual

1

Overview

1

Overview

Note: A *network* is a connection between a group of machines allowing them to transmit information to one another. A *local network* is the network surrounding your machine, whereas a *remote network* is a network that doesn't directly include your machine.

Electronic mail and *electronic messages* ease communication in the workplace. Many people find it useful to send and receive electronic mail and messages through a computer network, especially when it is impossible or inconvenient to communicate in person, by phone, or by post office mail.

Sending *electronic mail* is like sending a telegram. You can read, save, and edit electronic mail when you receive it on your machine.

Initiating an *electronic message* is more like calling someone up on the phone than sending a telegram, because it is immediate and often interactive. In other words, when you send an electronic message, the person who receives the message can read the message and reply to it while you wait. You can have a *synchronous* electronic conversation; in other words, both parties sending and receiving messages simultaneously through the computer network without their messages interfering with one another.

1.1. What Are Mail Tool and `mail`?

`mail` and Mail Tool are the programs you can use to send electronic mail. Mail Tool is a window interface to `mail`.[1] Because Mail Tool doesn't require you to memorize any specialized commands, we introduce it before `mail`. However, not everyone has a terminal which can run Mail Tool. In any case, the sections on `mail` and Mail Tool are independent of each other.

An optional file in your home directory, called the `.mailrc` file, alters the behavior of Mail Tool and `mail`. See Appendix A for more information.

1.2. Addressing Other Users

Every person running SunOS has a *username*; that is, a name by which he or she is known to the system. Each user also has a *machine name* associated with the machine he or she is using. You address mail or messages to people by using their usernames and machine names. There are three types of electronic addresses in SunOS; which one you use depends upon your relation to the person you're communicating with. Suppose that Betty Jo Bolinsky goes by the

[1] Historically, the `mail` and `Mail` programs differed considerably, but now they are the same. You can type either `mail` or `Mail` when you see examples that specify `mail`.

However, it *is* important that you have the right version of `mail`, the one that is located in the directory `/usr/ucb`. If you have another `mail`, such as the one in `/usr/bin`, in your path before `/usr/ucb/mail`, then Mail Tool will not work and `mail` itself will be different. See *Setting Up Your SunOS Environment: Beginner's Guide* on how to set your path.

username `bettyjo` and works on the machine `cupcakes`:

same machine
If you are on the same machine (`cupcakes`), then you only need to address her as

```
bettyjo
```

local network
If you're on different machines on the same local network, then you have to give her username and machinename, separated by the @ symbol:

```
bettyjo@cupcakes
```

(the @ is pronounced "at," making her "bettyjo at cupcakes").[2] However, on local networks using the Yellow Pages to maintain a database of mail aliases, the username *bettyjo* should suffice as a mailing address. This is because the Yellow Pages (set up that way) maintains a list of who's on which machine. Ask your System Administrator if your system is running the Yellow Pages.

In this manual, we have generally used both the username and the machine name. This is a good practice to get into, since it lessens the chance that you'll send something to the wrong person, and also because it saves the mail and message programs from looking on your machine for `bettyjo` when she's on another machine.

remote network
If you are on different networks, then you must use an addressing protocol unique to that network. For example, this is what Bettyjo's address might look like if you were communicating over UUCP or ARPANET networks, respectively:

```
neptune!bluetick!cupcakes!bettyjo
```

or

```
bettyjo@cupcakes.reed.EDU
```

Chapter 6 gives a brief introduction to sending mail over remote networks; for information on networks in general, see the *Using the Network: Beginner's Guide* manual.

[2] Occasionally someone may store his or her mailbox on a machine other than his or her own. In that case, give the name of the machine the mailbox is on.

1.3. Electronic Messages

Unlike electronic mail, electronic messages appear immediately on the receiver's terminal. There are three types of electronic messages:

interactive
The `write` and `talk` programs allow you to have a conversation with other users.

broadcast
The `wall` program allows you to broadcast messages to other users on your machine.

system
SunOS or its associated programs send messages to your *console*, such as error messages, or status information.

Note: The *console* is the entire screen, or a special window on the screen, where system messages appear.

Electronic messages use the same address schemes as electronic mail; however, most remote networks do not support interactive message programs like `talk` and `write`.

1.4. Suppressing Messages and Mail Notifications

When you want to stop display of `write` and `talk` messages to your console, type

```
mesg n
```

(`n` stands for "no"). If you're running the SunView windows system, then you must be *superuser* (also known as `root`) for `mesg` to work, and you must type it for every window in which you want to shut messages off. The manual *Doing More with SunOS: Beginner's Guide* explains how to become superuser. Another way to get `mesg n` to work is to put it on a line by itself in your `.login` file; `.login` is consulted every time you log in on your machine. *Setting Up Your SunOS Environment: Beginner's Guide* explains `.login`.

`biff`

Normally, you're not immediately notified of new mail; instead, the system waits until you finish what you're doing and type a new command. However, with the command

```
biff y
```

the system displays mail notifications on the terminal immediately: these notifications include the first few lines of the letter. By default, the system suppresses immediate mail notification (with `biff n`). As with `mesg`, you must be superuser to use `biff` if you're using SunView. You can also put `biff y` in your `.login` file. (For more on your `.login` file, see *Setting Up Your SunOS Environment: Beginner's Guide* .)

For more information on `mesg` and `biff`, see the *SunOS Reference Manual* or type `man biff` or `man mesg`.

This manual contains more information on remote and local networks, addressing conventions, and the like. See also *Using the Network: Beginner's Guide*, as well as the following entries in the *SunOS Reference Manual*: *mail*, *mailtool*, and

aliases.[3]

1.5. Other Useful Commands

A variety of other commands aid you in reading mail and sending messages. Further descriptions of these commands appear later in this manual.

`from` — Tells you who the mail in your system mailbox is from

`users` — Lists the username for each user currently logged in on the local machine

`who` — Lists the username, "terminal," and login time for each user currently logged in on the local machine

`w` — Lists the username, "terminal," login time, and other statistics for each user currently logged in on the local machine

`rsh` *machine-name command*
Executes *command* on machine *machine-name*. Useful for listing usernames, and other information about people on other machines.

`vacation` — Sends an automatic, pre-written letter in reply to mail when you're away.

Additionally, the `.forward` file, described later, allows you to forward your mail to other users, machines and programs.

[3] You can get information on these commands on-line on your computer by typing `man` followed by the entry you want. (Note that Mail Tool is spelled *mailtool*.)

2

Mail Tool

2

Mail Tool

2.1. What is Mail Tool?

To avoid confusion, we will refer to electronic mail messages as *letters*. In any case, mail messages should not be confused with the messages referred to in the "Messages" chapter of this manual.

Mail Tool is a window-based program which makes it easy to use SunOS's mail-handling facilities. With Mail Tool you can write, send, and receive electronic mail to and from other SunOS or UNIX system users, even if they are physically far away from you, and even they're not using a Sun system.[4] Mail Tool is actually built on top of an existing SunOS program called simply `mail`, which is described later on in this manual.[5] `mail` is not windows-based; it doesn't have any graphical layout, as Mail Tool does. To use `mail`, you need to know a number of commands to type in, whereas Mail Tool allows you to pick commands from a menu or with buttons, as you would from a vending machine.

Who Can Use Mail Tool?

Because of its ease of use, we are placing this chapter on Mail Tool ahead of the chapter on `mail`. Nonetheless, not everyone will be able to use Mail Tool. To run Mail Tool, you must have a bit-mapped screen, like the Sun Workstation. And you must be using the SunView windows system. (If you're not sure whether you have a bit-mapped screen, ask your system administrator.)

And **please note**: we are assuming for this section on Mail Tool that you are already familiar with the material in the *SunView 1 Beginner's Guide*, so that terms like "pushing a button," "scrolling," "pull-right menus," and "cycle items" will not be completely mysterious to you. If for some reason you can't read that manual, yet still want to send and receive electronic mail, you should probably start with `mail` instead. You can always switch to Mail Tool later.

2.2. Starting Mail Tool

You must be running SunView before you can use Mail Tool. (See the *SunView 1 Beginner's Guide* for information on how to run SunView.)

There are two ways to start (or "invoke") Mail Tool:

Starting Mail Tool from a Command Line

One way to start Mail Tool is by typing `mailtool &`. It should look something like the following:

---

[4] In fact, they do not have to be using a UNIX-based system.

[5] Mail Tool looks for `Mail` when it starts up; `Mail` and `mail` should be the same thing.

Figure 2-1 *Starting Mail Tool on a Command Line*

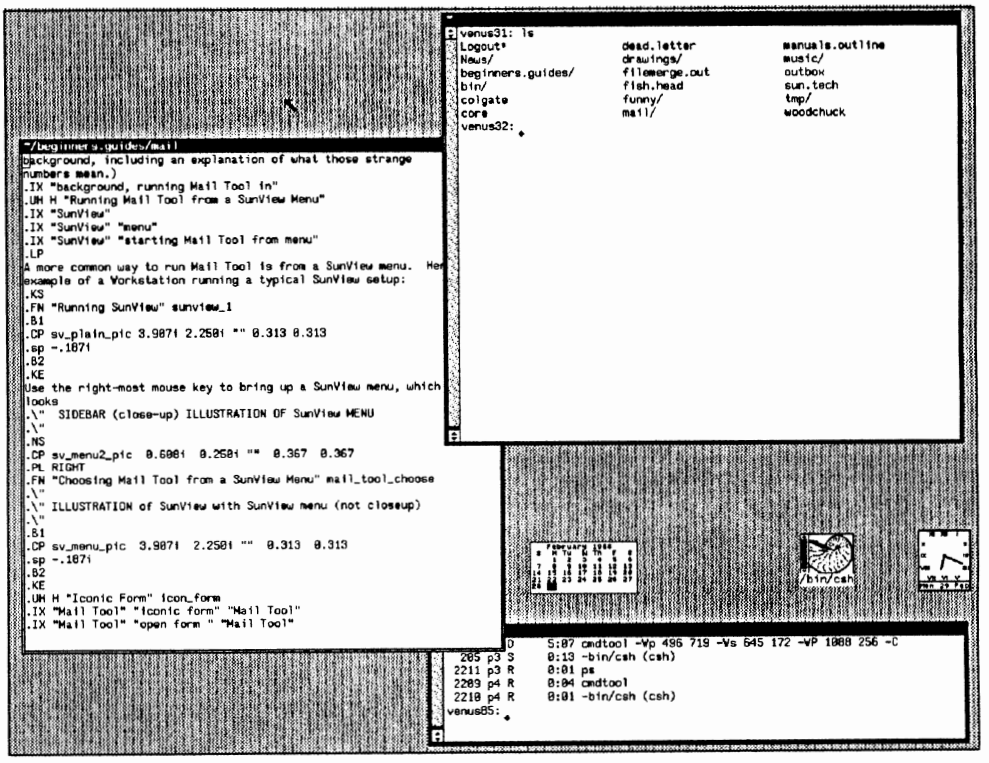
```
venus% mailtool &
[1] 68022
venus%
```

The & is optional. It tells Mail Tool to run in the *background*, so you can still do other work in the same window while using Mail Tool. (See the *Getting Started with SunOS: Beginner's Guide* for more information on doing things in the background, including an explanation of what those strange numbers mean.)

Running Mail Tool from a SunView Menu

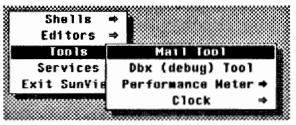
A more common way to run Mail Tool is from a SunView menu. Here's an example of a Workstation running a typical SunView setup:

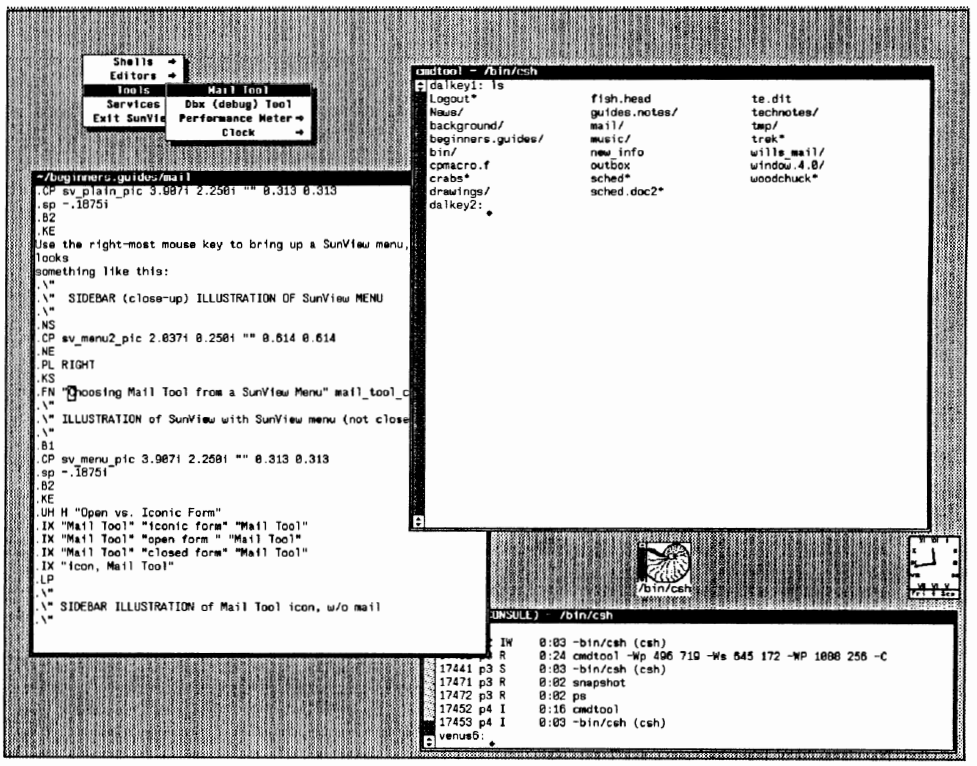
Figure 2-2 *Running SunView*



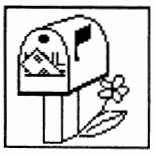
Use the right-most mouse key to bring up a SunView menu, which looks something like this:

Figure 2-3 *Choosing Mail Tool from a SunView Menu*





Iconic Form



When Mail Tool comes up, it appears in *iconic* form; ie., it comes up as a little box, the *icon*, which represents the running Mail Tool program.

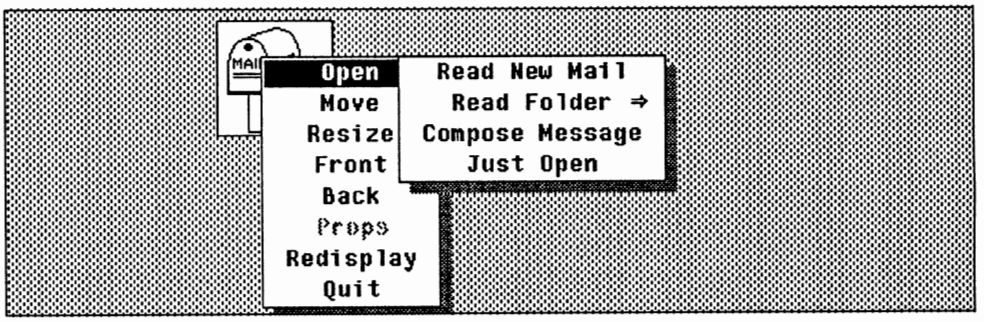
When Mail Tool is displayed in this fashion, it is said to be running *closed*. When you receive mail from someone, the Mail Tool icon changes: the little flag on the mailbox goes up, and a letter appears in the mailbox's slot.

Opening Mail Tool

The purpose of the icon is to let you run Mail Tool without taking up much room. To *use* Mail Tool, though, you have to open it. The *SunView 1 Beginner's Guide* describes various methods of opening an icon such as Mail Tool's; one such method is to use the mouse to move the pointer on the screen into the Mail Tool icon, and then click the left mouse button.

Like most window icons, the Mail Tool icon has a menu which allows you to manipulate the application in various ways: moving it, resizing, quitting it, and so forth. You can also open Mail Tool with this menu. The 'Open' menu item is a pull-right item, which itself offers the following ways of opening Mail Tool:

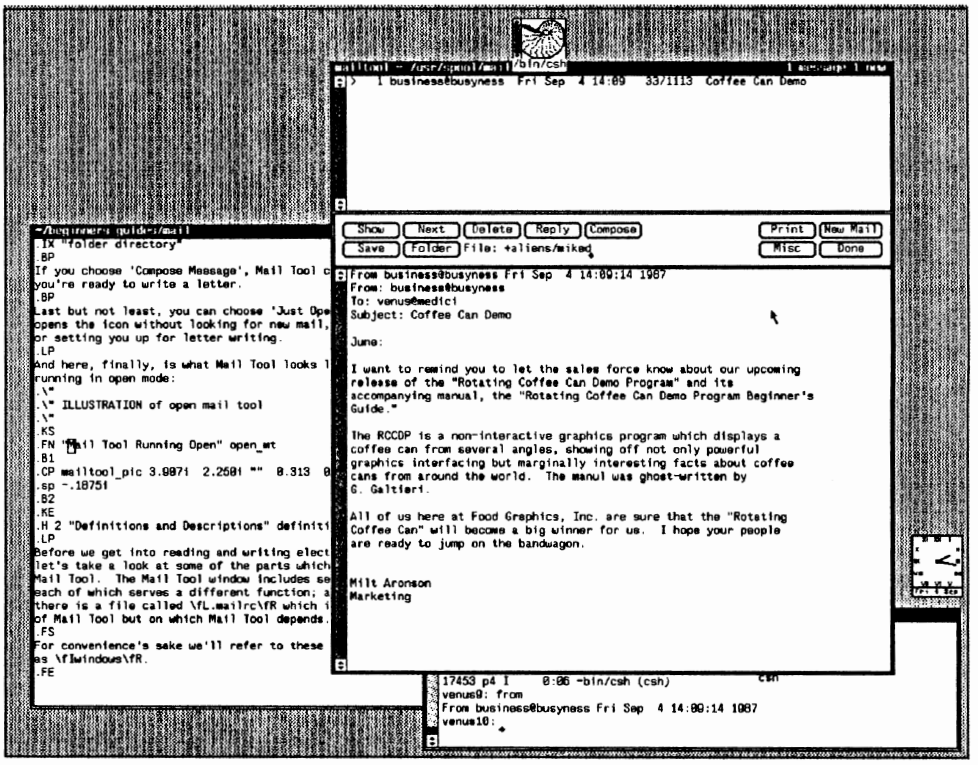
Figure 2-4 *The Iconic Open Menu*



- The first menu item is 'Read New Mail'. When you open Mail Tool this way, it automatically checks for new mail.
- The next item in the 'Open' menu is 'Read Folder'. A folder is a kind of a file which contains letters; it's explained in Section 2.10. 'Read Folder' is itself a pull-right; pulling it displays all your folders. You can then pick the folder you want to look at.
- If you choose 'Compose Message', Mail Tool comes up so that you're ready to write a letter.
- Last but not least, you can choose 'Just Open', which opens the icon without looking for new mail, reading in a folder, or setting you up for letter writing.

And here, finally, is what Mail Tool looks like when it's running in open mode:

Figure 2-5 *Mail Tool Running Open*



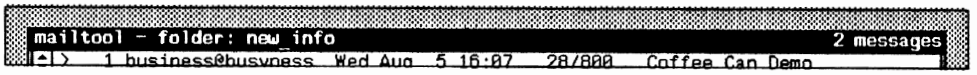
2.3. Definitions and Descriptions

Before we get into reading and writing electronic letters, let's take a look at some of the parts which make up Mail Tool. The Mail Tool window includes several subwindows,[6] each of which serves a different function; additionally, there is a file called `.mailrc` which is not part of Mail Tool but on which Mail Tool depends.

The Frame Header

The *frame header* is the stripe at the top of Mail Tool, and it displays information about what you're doing — what file or folder you're working on, how many letters you're working with, and so on. This is what a typical frame header might display:

Figure 2-6 *The Frame Header*



In figure 2-6 the frame header tells you that you are working with the folder *new_info* which contains 2 letters. Other examples of frame header messages are "No mail," "retrieving new mail," "building folder menu," and the number of

[6] For convenience's sake we'll refer to these subwindows as *windows*.

letters you've deleted. You use the frame header to keep track of whether you've got new mail, how many letters you have to read, how many letters you've gotten rid of, and so on.

The Header List Window

The *header list window* is the box just below the frame header. It displays information about your *mailbox* (also known as the *system mailbox*). Your mailbox is a file containing the letters you receive; its name is displayed in the frame header when you look at new mail. The mailbox is generally a file with your username located in the directory `/usr/spool/mail` — if your username is *willie*, your mailbox is `/usr/spool/mail/willie`.[7]

Note: the word "header" as used here has no connection with the "frame header."

The letters in your mailbox are summarized in the *header list*. Each letter has a one-line entry in the header list called a *header summary.*

Here's a picture of the header list window:

Figure 2-7 *The Header List Window*

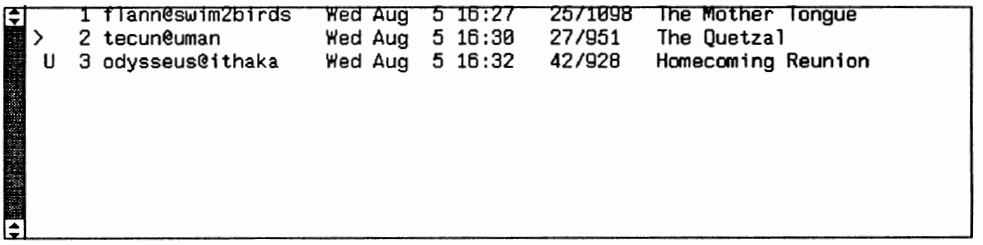


Figure 2-7 shows a header list with three header summaries. The first summary is of a letter from user *flann* on the machine *swim2birds*; the second letter is from someone with the username *tecun* on machine *uman*, while the third letter is from *odysseus*.[8]

If you're familiar with `mail`, you'll recognize these header summaries as being identical to those in `mail`.

The letters referred to in the header list may be ones that you have received recently, or they may be mail which you received a long time ago but never did anything with (like filing them or throwing them away), or they may be letters which you've stored away in a file or folder. (We'll talk about folders in Section 2.10.)

Each header summary contains the following information:

status msg. no. sender date time msg. length subject

- There are three *status indicators* for a mail letter. An *N* means the letter is *new*. A *U* indicates that the letter is *unread*; you've retrieved it from your

[7] Actually, your mailbox is probably located in `/var/spool/mail` instead of `/usr/spool/mail`; however, SunOS maintains a symbolic link from the latter for compatibility with older versions of mailing programs.

[8] Flann O'Brien, alias Myles Na Gapoleen, was the author of *At Swim-Two Birds*. Tecun Uman was a famous Mayan warrior chieftain at the time of the Spanish conquest.

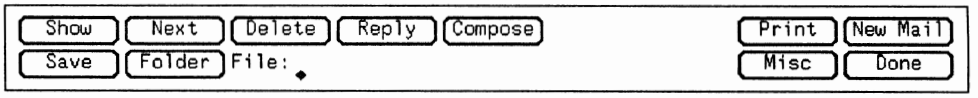
mail box—and possibly stored it away—without reading it. The last status indicator is nothing at all; a blank here just means that you've retrieved and read the mail, but you haven't done anything with it. The > indicates the letter you're currently looking at.

- Each letter has a *letter number* assigned to it in the order it's received.
- The *sender* is the person who sent you the mail.
- The *date* is the date he or she sent it.
- As you'd expect, *time* is the time the message was sent.
- The *letter length* is given in two parts: the first number is the number of lines in the letter, and the second number is the number of characters.[9]
- The *subject* is whatever the sender says it is.

The Command Panel Window

This is the *Command Panel window*:

Figure 2-8 *Command Panel Window*



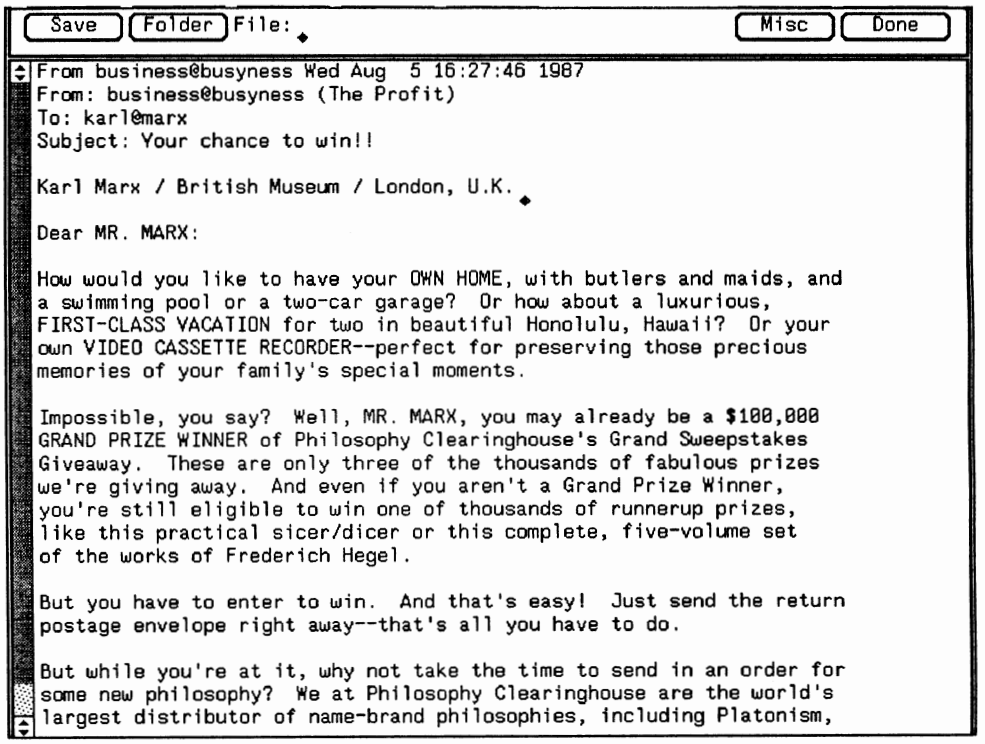
Each of the buttons will be explained later, along with the `File:` text item.

The Message Window

The window beneath the command panel window is called the *message window* because it's here that Mail Tool displays letters that you receive:

[9] You can choose to have Mail Tool hide some header lines when it displays a letter, so sometimes you won't see all the lines and characters the letter has. See Appendix A and Section 2.8.

Figure 2-9 *The Message Window*

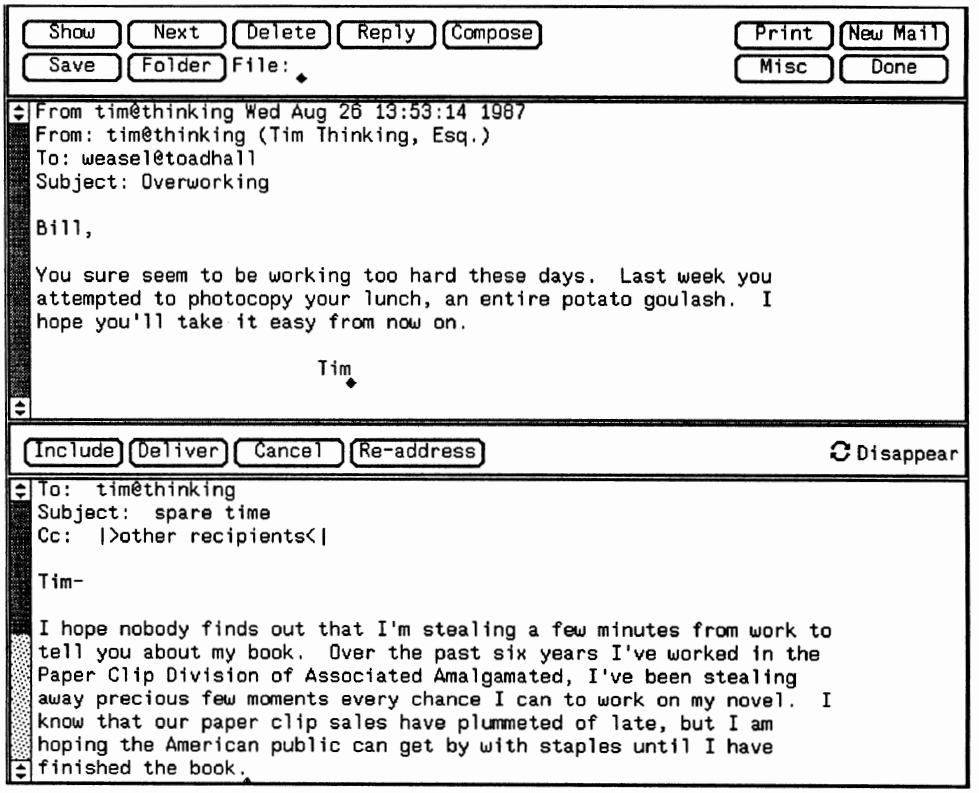


You may have noticed in Figure 2-9 that the body of the letter itself is preceded by pretty much the same information we saw in the header list window: who sent the letter, when, its subject, and so on. This information makes up the letter's *header*. (A header summary, then, is a shortened version of a header.)

The Composition Window

The composition window is where you write letters. Here's what the composition window looks like when you bring it up from Mail Tool:

Figure 2-10 *The Composition Window*



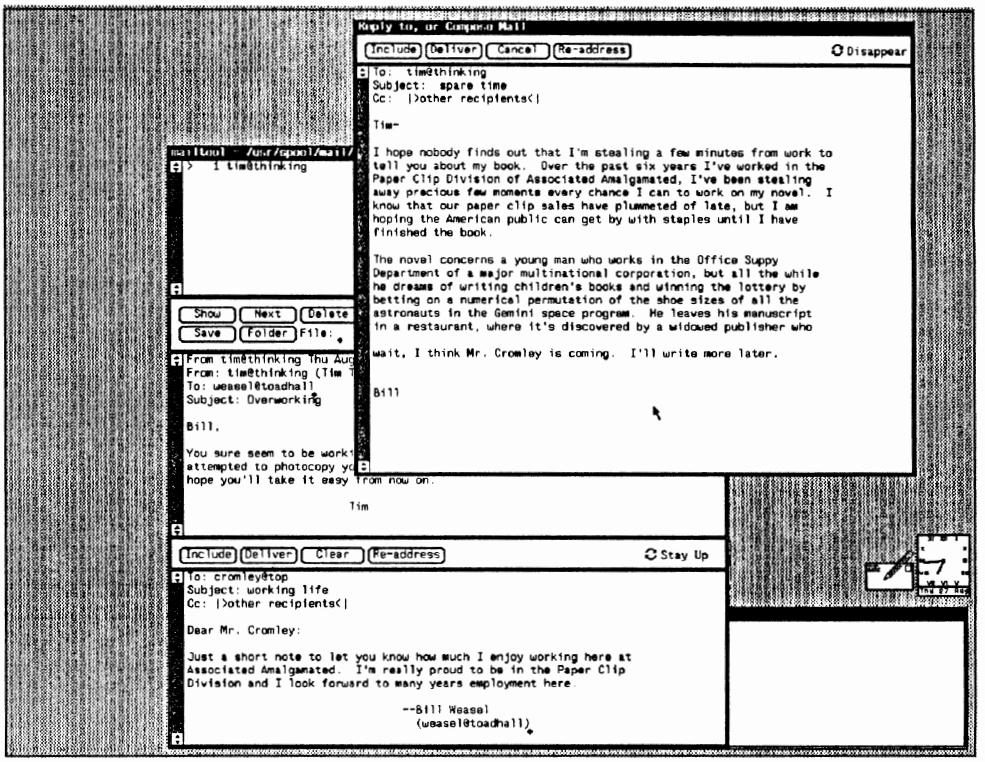
You can bring up more than one composition window at once, enabling you to write several letters at a time. As in Figure 2-10, the first composition window is normally created by splitting the message window in two.[10] Subsequent composition windows, however, come up as separate, or *pop-up* windows. You have to be in Mail Tool to start such a window, but once you've started it, you can move it, close it, resize it, etc., just like any other window. In fact, you can even close Mail Tool and still write a letter in a pop-up composition window. (Conversely, you can close the composition window and still run Mail Tool in its open mode.) What's more, you can run several composition windows at once, whether open or closed.

You can alter Mail Tool so that the first composition window also comes up as a pop-up window. This is explained in Section 2.13 and Appendix A.

Here's another example of using the composition window. In this case we're running three composition windows: one is split off from the message window, one is partially covering Mail Tool, and one is closed down to its icon (a pencil poised over an envelope):

[10] Technically, a composition window which is brought up this way is a subwindow, not a window.

Figure 2-11 *Another Exciting Letter-Writing Session*



We discuss bringing up the composition window this way in Section 2.13 and Appendix A.

The `.mailrc` File and The Defaults Editor

Before we go on to reading and writing letters, let's mention in passing the `.mailrc` file and the Defaults Editor. The `.mailrc` file contains various settings that determine the way Mail Tool looks and runs — things like how often Mail Tool should look for new mail, how big the composition window should be, where Mail Tool should store your letters, and so on.

The Defaults Editor is a SunView program which allows you to change settings for various Sun tools, including Mail Tool. You can use the Defaults Editor to modify the `.mailrc` file in a simple and rapid way. The Defaults Editor is explained in the *SunView 1 Beginner's Guide*; Appendix A is a list of the Mail Tool settings you can change.

2.4. Reading Mail

Incoming Mail

There are a couple of ways that you're notified of incoming mail. First, outside of Mail Tool, SunOS will give you the notification

```
You have new mail.
```

in the window you're typing in. Second, when Mail Tool is in its closed form, the icon changes, as we noted earlier in Section 2.2. Third, when you're running Mail Tool in its open form, the frame header at the top of Mail Tool will display

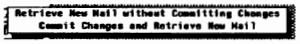
"[New Mail]" when you receive a letter. Additionally, you can use the Defaults Editor to make Mail Tool beep and flash the screen when it receives a letter. (Mail Tool beeps and flashes in both iconic and open mode.) See Appendix A.

As noted in Chapter 1, the `biff y` command gives you *immediate* notification of new mail, including a short excerpt from the new letter. This is useful if you tend to answer your mail right away. Many users keep this immediate notification turned off, since they find it a nuisance to be interrupted every time they receive a mail letter.[11]

2.5. The New Mail Button

Clicking the left mouse button on a Mail Tool button always selects the topmost menu item for that button.

The **New Mail** button does just what you'd think it does: it retrieves letters that you've received. Like the other buttons in the command panel window, you *push* **New Mail** with the leftmost mouse key. Use the right mouse key to get the menu associated with the button, and to choose from it.



The **New Mail** button has two different menus, depending on whether you are currently looking at your mailbox, or at a file or folder. When you are looking at your mailbox, you see the menu at left.

Committing Changes

The difference between the two menu items reveals one of the more important concepts about Mail Tool: *committing changes*. When using Mail Tool you'll want to delete letters from time to time, letters that you don't need to store anyplace. When you delete them, they are removed from the header list. However, they stay in your mailbox, and you can later recover, or "undelete" them. Doing a commit, though, removes deleted letters from your mailbox, making your changes permanent.[12]

There are three places where Mail Tool automatically performs commits. One is when you end a Mail Tool session, either closing Mail Tool to its iconic state or ending Mail Tool altogether; another is when you retrieve a file or folder; the last is when you're retrieving new mail. We'll talk about the first case when we discuss the **Done** button and the second when we look at the **Folder** button; let's look at the situation for new mail now.

If you push the **New Mail** button while looking at your mailbox, Mail Tool retrieves new letters *without doing a commit*; you can still undelete letters you've deleted. On the other hand, if you choose the 'Commit Changes and Retrieve New Mail' menu item, you do a commit and all your deletions become permanent.

Note: if you are looking at a file or folder, then the **New Mail** menu changes. Pushing **New Mail** *always* does a commit when you are looking at folders or files. See Section 2.10 for more on folders.

---

[11] Normally `biff` is turned off; if you turn it on, you won't see the `You have new mail` message.

[12] If you have the `hold` variable in your `.mailrc` file turned off, then a commit moves already-read letters from your mailbox to a storage place called 'mbox'. See Appendix A and Section 2.16 for more information.

2.6. Choosing Letters

When you retrieve new mail, Mail Tool displays an updated header list in the header list window, and it displays the current letter in the message window.

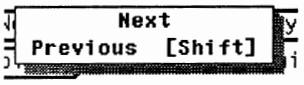
If you have unread letters when you push the **New Mail** button, you'll point past them to the new letter. Mail Tool will mark them with a `U` as unread, however.

Just what is the current letter, though? When you push **New Mail**, Mail Tool points to the first new letter in your mailbox. (By "points to" we mean that Mail Tool puts that letter in the message window and puts a ">" in front of its header summary to indicate that it's the letter you're looking at.)

Note: You can choose a letter from the header list just by moving the mouse pointer to its header summary and clicking the left mouse button. This selects that letter as the *active* letter, or the one that most panel buttons will affect if pushed.[13] Thus, in most cases, you can do an operation on a letter just by clicking on its header summary and pushing the panel button you want, without having to display the letter.

To read other new mail, you can move through the header list with two buttons, **Next** and **Show**.

2.7. The Next Button



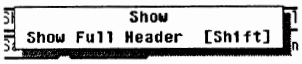
The **Next** button moves you to the following letter in the header list. You can also look at the letter that comes before the current letter; you do this by either holding down the Shift key when you push the **Next** button, or by choosing 'Previous' from the **Next** menu. If you're looking at the last letter in the letter list, pushing the **Next** button causes Mail Tool to display the previous letter.

2.8. The Show Button

Another way to move around in the header list is by using the **Show** button. To display a particular letter, move the pointer on the screen to the header summary you want. Click the leftmost mouse button to choose that header summary; then push the **Show** button. The letter you've chosen will appear in the message window.

Displaying the Full Header

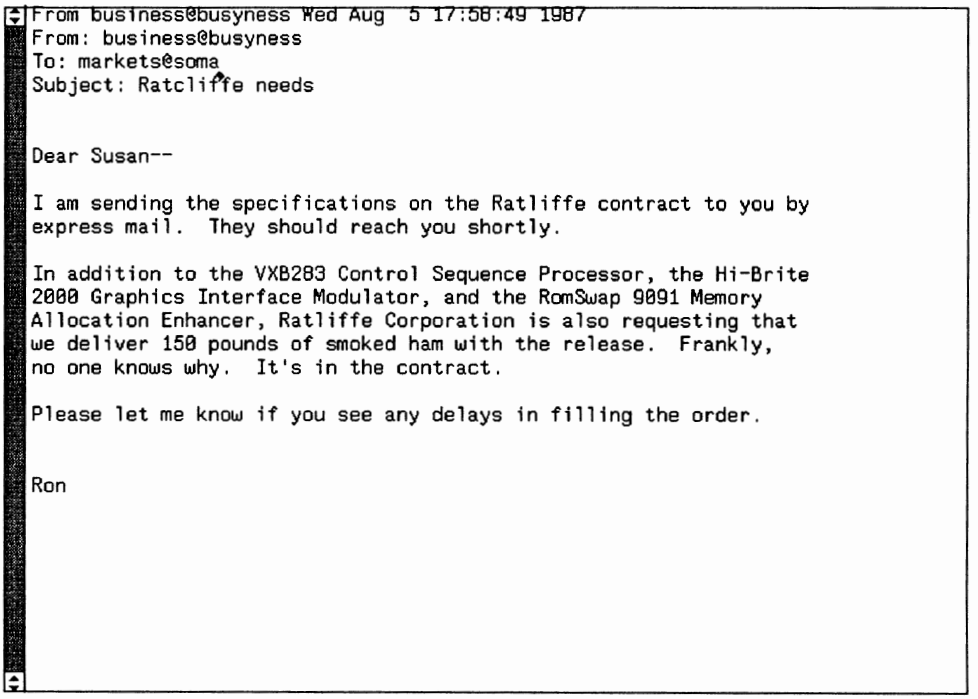
Because a letter's header can contain quite a bit of information, you can change `.mailrc` to have Mail Tool suppress unwanted information in its display. (This is explained in Section A.3 of Appendix A.) This way, although the information is still there, your letters display only things you're interested in.



But occasionally you may have a reason to want to see all the information in a letter's header. For example, you may want to know the full path the letter took to reach your machine. Or you may have a need to know the letter's ID number. For that reason the **Show** button has a 'Show Full Header' menu item (which you can get by using the Shift key on the **Show** button). Here is the same letter, first displayed normally, and then with the 'Show Full Header' option:

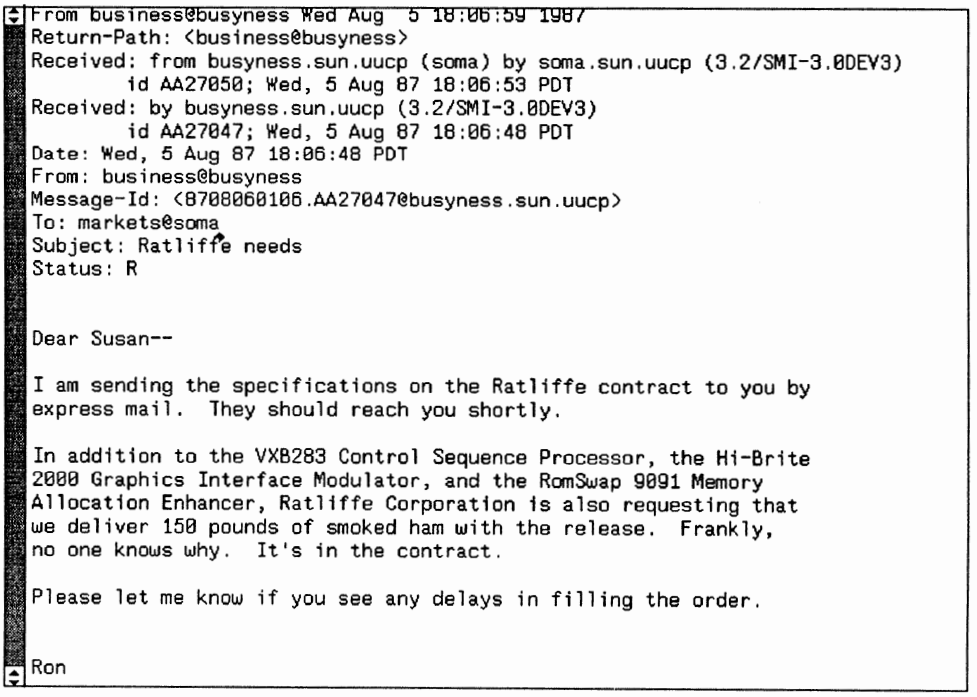
[13] **Next** is an exception.

Figure 2-12 *A Typical Mail Header*



Here's the same letter with the full header displayed:

Figure 2-13 *Full Letter Header*

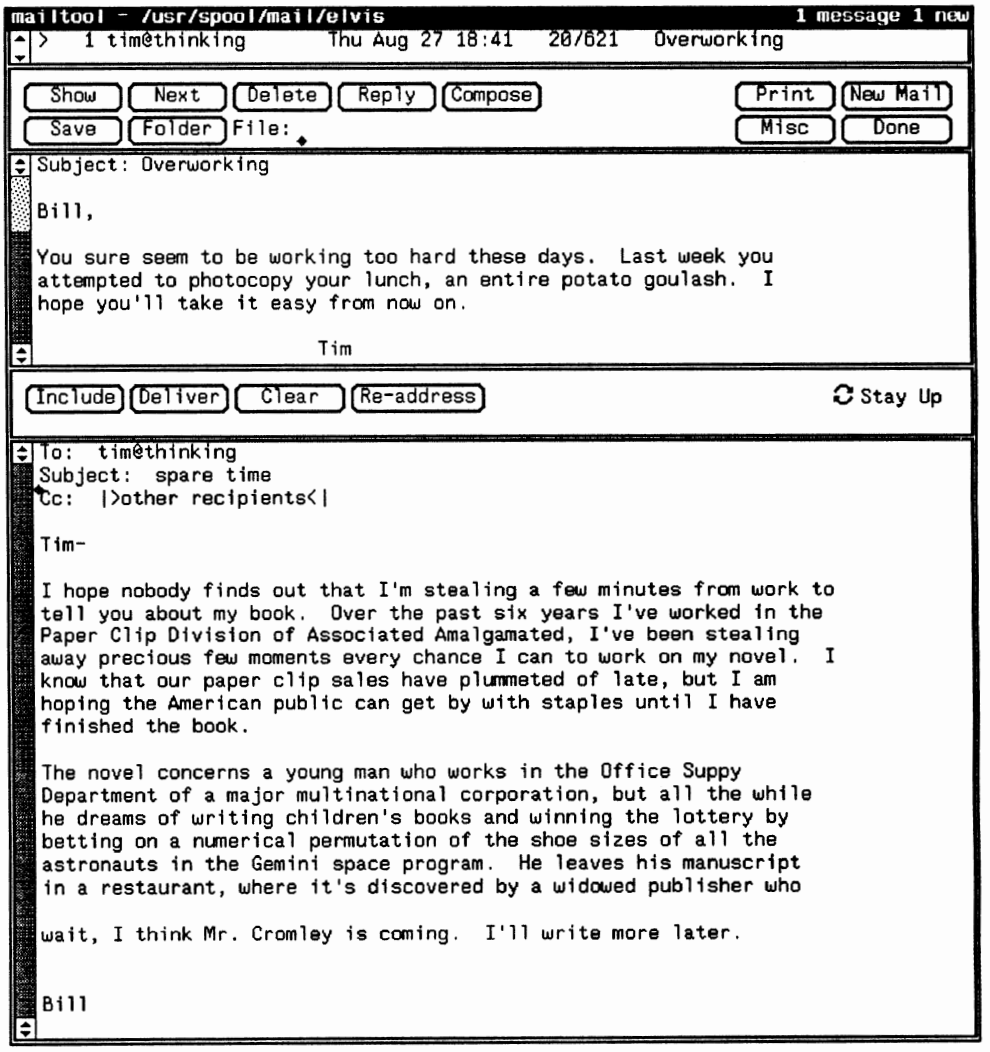


Scrolling Through the Header List Window

If you get a lot of mail letters, your header list may get too long for all your headers to be displayed at one time. Like other SunView windows, you can scroll through the header list window by using the scroll buttons on its left-hand side. You can also split the window into two or more sections for multiple views. Splitting and scrolling are explained in the *SunView 1 Beginner's Guide.*

As a matter of fact, you can treat your Mail Tool windows the way you treat other SunView windows: you can move and resize them as you like. For example, here's Mail Tool with the various windows resized:

Figure 2-14 *Resizing Mail Tool Windows*



Resizing windows is explained in the *SunView 1 Beginner's Guide.*

2.9. The Print Button

Use the **Print** button to print out copies of your letters on a hard-copy printer. (You can change the way you print mail out by changing the `.mailrc` variable `printmail`; see Appendix A.)

2.10. Files and Folders

Of course, it isn't sufficient just to compose and read mail. You also want to be able to save and retrieve letters. You've probably noticed that in the command panel window there's a little item which looks like this:

```
File:
```

You type the name of a file to contain letters after `File:` For example, you might put all the letters you receive about recipes into a file called `recipes_mail` and all your letters about South American rug-cleaning fiction into `rug_fiction`. Such files can contain a large number of letters, and you can add, subtract, and view the letters they contain at any time you like.

Suppose you're saving a letter about Rodolfo Miramar's classic eighteenth-century novel *How Green Was My Carpet* into the file `rug_fiction`. If you just type

`rug_fiction`

after `File:` Mail Tool will assume that `rug_fiction` is located in its current directory.[14] If you want `rug_fiction` to be in a different directory, say `/home/medici/mail`, you can type in the file's whole name, `/home/medici/mail/rug_fiction`, after `File:`

Folders

But to avoid all that typing, Mail Tool (and `mail`) have *folders*. A folder is a file like any other, except for one thing: it's located in a place called a *folder directory* , which you designate. And a folder directory is useful because you can use a plus sign (+) as a sort of shorthand, an abbreviation, to designate it. For example, if you've designated `/home/medici/mail` as your folder directory, then `+rug_fiction` is interpreted by Mail Tool as `/home/medici/mail/rug_fiction`. That means that any time you want to use a folder, whether to store or retrieve a letter, you need only type in a + and the name of the file you're working with. (Together the + and the filename constitute the folder's name.)

You specify your folder directory by setting the `folder` variable in your `.mailrc` file. You can do this with an editor like `vi`, or, preferably, with the Defaults Editor. Appendix A explains how to do this.

Remember that folders are just files. The + is invisible outside of `mail` and Mail Tool.

Here's an analogy: if you work in an office, you might need to file away memos into files which go into different cabinets. You mark a "+" on all the files which go into the green cabinet; that way you don't have to write "This goes into the green cabinet" on all your files of memos. The memos and the files which contain them are unchanged by the "+" sign, and you always know which cabinet to look in to find memos, whether they're about South American rug fiction or

[14] Whatever directory Mail Tool was started from. Oftentimes this is your home directory. You can find out what the current directory is, and change it, with the Misc button. See Section 2.16.

recipes.

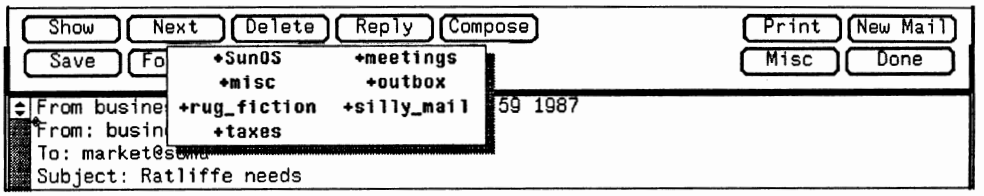
Just so you understand: you don't *have* to put letters in your folder directory. You can store a letter into a file anywhere you want, just by typing that file's full pathname in after `File:` (In our analogy above, you can still put files into the blue cabinet by writing "blue cabinet" on them, instead of using the plus sign.) But besides saving yourself a lot of typing, using folders tends to keep your letter files in one place, so that you always know where to look for them. And the **Folder** button, described next, makes things even easier.

The Folder Button

Now that you know what a folder is, let's look at its namesake, the **Folder** button. This is the button you use to do two things: **a**) *select* the name of a folder to use, and **b**) *retrieve* a folder, or any file you specify.

By clicking the rightmost mouse button over the **Folder** button you get a menu of all the folders in your folder directory. You then choose the one you want; the name of that folder now appears after `File:`[15] This is often faster than typing the folder's name after `File:`

Figure 2-15 *A Folder Directory*



The other thing you do with the **Folder** button is retrieve a file or folder with it. (This is also known as "reading a file, or folder, in.") Pushing the **Folder** button with the leftmost mouse key selects the file or folder whose name appears after `File:` It also commits all your changes.

The **Folder** button is really somewhat misnamed, because it will retrieve *any* mail file, whether or not that file is a folder.

It's important to recognize that selecting a folder's name *does not* mean you are yet working with that folder. You don't actually switch to the folder whose name you've chosen until you retrieve the folder with **Folder**.

Let's review how you would work with a file or folder:

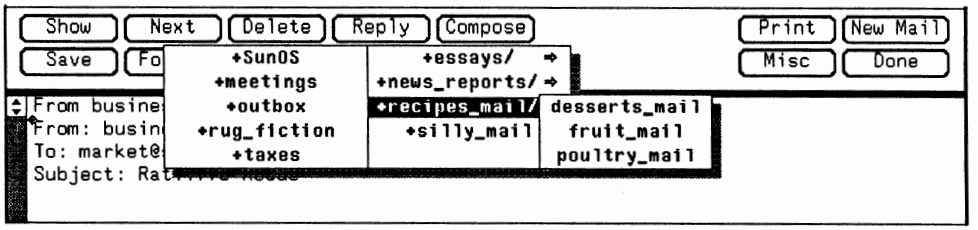
- If you're saving a letter to a new file or folder, you type it's name in after `File:` You then save it with the **Save** button, described later on.
- If you're working with an existing folder, you use the **Folder** button to put its name after `File` (You can also type its name there.) You can then save to it with **Save**, or retrieve it with the **Folder** button.

[15] Mail Tool will display every file in your folder directory, whether or not it contains letters. But Mail Tool will become confused if you ask it to retrieve a non-mail file; for this reason it's a good idea to keep only letter files in your folder directory.

- If you're working with an existing file outside your folder directory, you type its name in after `File:` Like a folder, you use **Folder** to retrieve it. You can also save to it with **Save**.

Folders can be directories as well as being files. That means that you can have a subdirectory of folders in your folder directory. For example, suppose you had your recipe letters divided up into folders containing letters about poultry, desserts, and fruit. You could make `recipes_mail` a directory and have folders in it called `poultry_mail`, `desserts_mail`, and `fruit_mail`. When you use the **Folder** button to look at your folder directory, `recipes_mail` appears with a ⇒ to indicate that it is a pull-right menu containing other items, in this case the folders `poultry_mail`, `desserts_mail`, and `fruit_mail`.

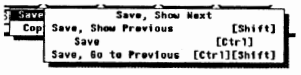
Figure 2-16 *Directories of Folders*



Note: from now on we will refer to files and folders as simply "folders," unless the distinction is important.

In addition to being files and directories, folders can likewise be *links* to other files and directories. That way, for example, you can share a folder with another user. See the manual *Doing More with SunOS: Beginner's Guide* for more on links.

2.11. Saving Mail



Now that you know how folders work, it's time to look at saving mail with the **Save** button.

To save a letter into a folder, you must first have specified its name after `File:` The **Save** button has two major options: saving a letter and copying it. The difference between saving and copying is simply that saving *deletes* the letter, moving it into wherever you've specified, while copying leaves a version where you found it.[16] So if you save a letter from your mailbox to a folder, the letter gets deleted from your mailbox unless you use 'Copy'.

A Sample Save Session

Here's how you might go about getting a letter, saving it into the `+secrets` folder, and copying it into the folder `+plans`.

1. Press the **New Mail** button to get new mail.
2. Choose a folder name from the **Folder** menu. In this case, `+secrets`. (If `+secrets` doesn't already exist, type its name in after `File:`)

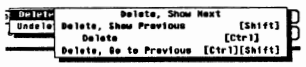
[16] You can undelete saved letters as though you'd used the **Delete** button; see Section 2.12.

3. Find the letter you want in the header list of new mail, and click on it. (You may want to **Show** it to make sure it's the right one.).
4. **Save** it to `+secrets`.
5. Retrieve the `+secrets` folder with the **Folder** button.
6. Find the letter you want to save in the header list, as in Step 3.
7. Choose the `+plans` name from the **Folder** menu, just like step 2.
8. Use the 'Copy' option of **Save** to put a copy of the letter into `+plans`.

(Can you think of a faster way to have done the foregoing?)[17]

One more thing about saving a letter into a folder: you can get a menu of just the folders you've switched to during this Mail Tool session. You get it by clicking the rightmost mouse button over `File:` If you work with a few folders over and over, this can save you time, since this menu is is often shorter than the **Folder** button menu of all the folders in your folder directory.

2.12. Deleting Letters



The **Delete** button removes letters from your mailbox or from folders. But because a deleted letter is actually kept, invisibly, in your mailbox or folder, a deleted letter can be recovered by using the 'Undelete' option of the **Delete** menu. As mentioned earlier, a letter can be undeleted at any time up until you commit changes (usually when getting new mail or finishing a Mail Tool session).

To delete a letter, find it in the header list and click the left mouse key to select it (as described in Section 2.8), then push **Delete**.

2.13. Composing and Sending Mail

At last we're ready to look at how to compose letters and send them. This covers the **Compose** and **Reply** buttons, as well as the composition window. Because the **Compose** and **Reply** buttons' menus are oriented around the composition window, we'll look at the window first. Then the button menus should be self-evident.

The Composition Window

The composition window is a SunView text window, and has menus and behavior associated with such windows. (Check the *SunView 1 Beginner's Guide* for information on text windows.)

Recall from Section 2.3 and Figures 2-10 and 2-11 that normally the first composition window comes up as part of the message window, and all subsequent composition windows come up as pop-up windows.[18] You can change Mail Tool so that the first composition window also comes up as a pop-up window by using the Defaults Editor to set the `.mailrc` variable `alwaysusepopup` to 'Yes'. To understand all this jargon, see Appendix A.

---

[17] You could have simply copied from your mailbox to `+secrets` and then saved from the mailbox to `+plans`, or you could have copied it to both places and then deleted it.

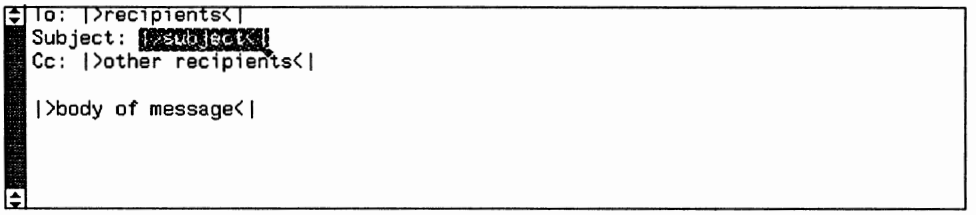
[18] There is a limit on the number of composition windows you can have running at once. If you exceed that limit, you'll get a warning which says "Not enough fds for any more pop-up windows."

When you bring a composition window up as a pop-up, you can open, close, resize, etc., it without affecting the rest of Mail Tool. You cannot create the composition window from outside Mail Tool, though, and it disappears when you quit Mail Tool.

Fields

The first thing you see when you look at a composition window is that it is "pre-addressed." There are lines which look like this:

Figure 2-17 *Subject and Address Fields in A Composition Window*



Each of the areas marked out by `|>` and `<|`, such as `|>recipient<|`, are *fields*. You type in appropriate information for each field; the darkened field is the *active field*, and it is replaced by your input. To move between fields, you type Ctrl-Tab (that is, you hold the Ctrl key down while typing the Tab key). An empty field is not sent as part of your letter.

(In fact, there's nothing magical about these fields; you can still type, erase, and use the pointer to move around, just as in any SunView text window. You can even set up Mail Tool to not present you with fields, by setting the `.mailrc` variable `disablefields`. See Appendix A.)

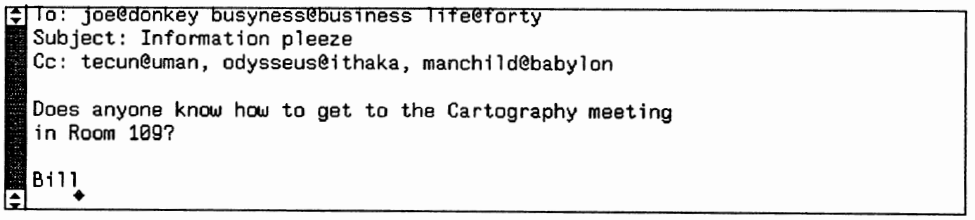
The Cc: Option

In Figure 2-17 there is a line which reads:

```
Cc:
```

This stands for *Carbon copy*, and you use it to send a copy of your letter to someone besides the addressee. You can Cc: more than one person, as long as the recipients' names are separated by spaces (with or without commas). This is true for the `To:` line as well:

Figure 2-18 *Multiple Addressees*

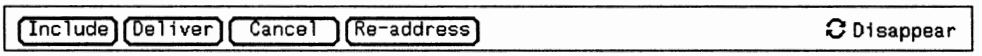


You can choose to have the composition window come up without the `Cc:` line, by choosing an item from the **Compose** menu which doesn't have a "Cc:" in it.

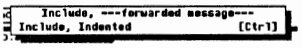
Composition Window Buttons

These are the buttons in the composition window:

Figure 2-19 *Composition Window Buttons*

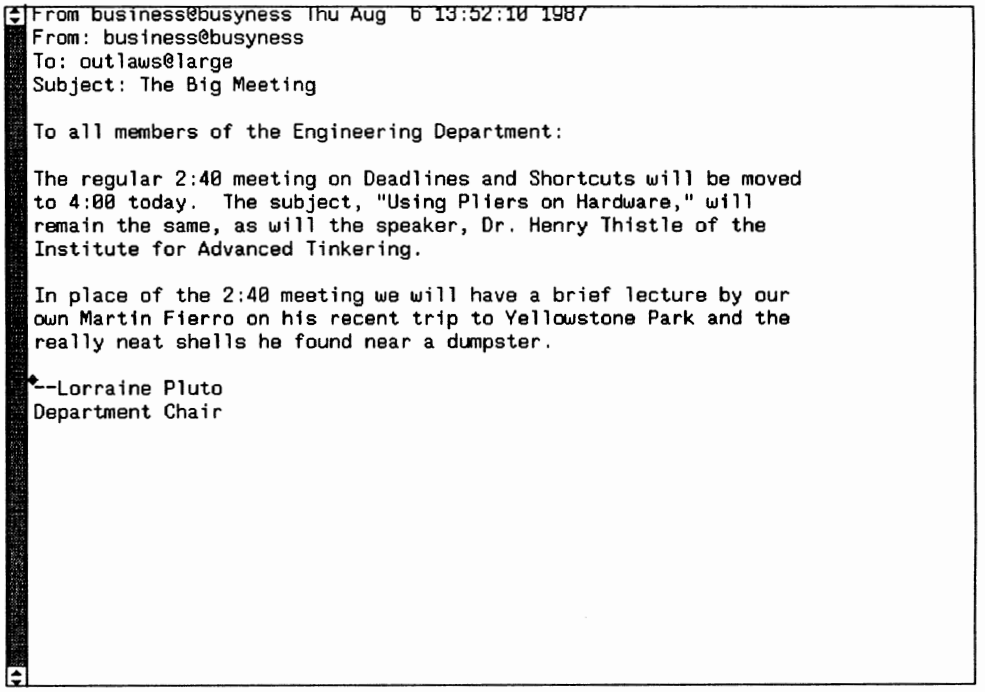


Including Letters



Sometimes you'll get a letter that you want someone else to see. Mail Tool allows you to forward letters you receive; you can use the **Include** button to put a copy of the letter into a letter you send. For example, suppose you receive this letter, and you want to send a copy of it to `joe@donkey`:

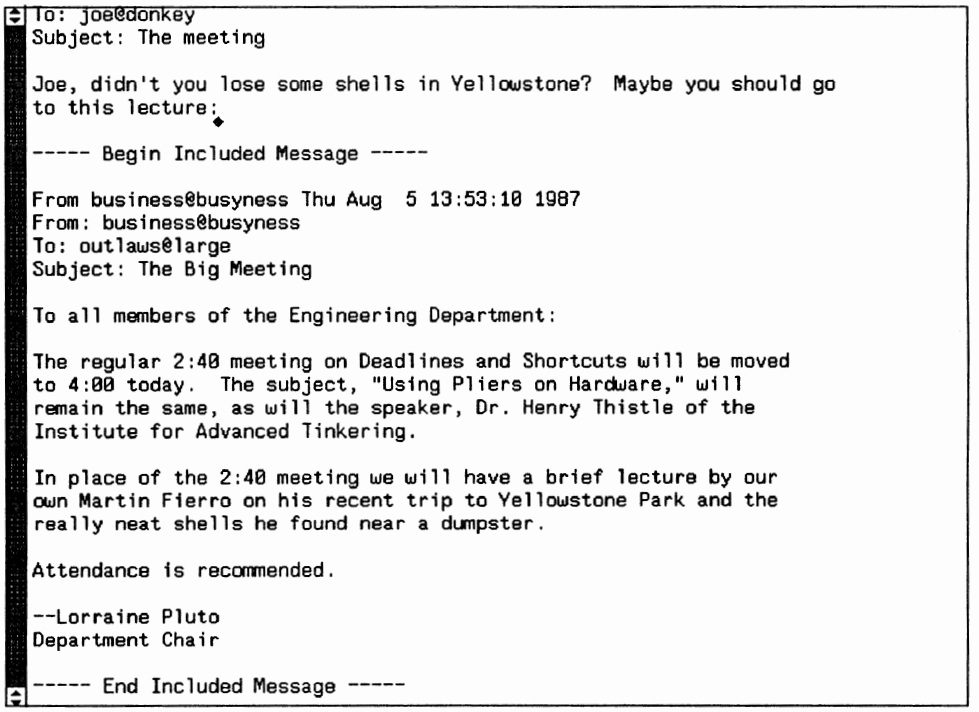
Figure 2-20 *A Received Letter*



First, you bring up the composition window with **Compose** or **Reply**. Then make sure that the caret in the composition window is at the point you want the inserted letter to go. (Often this means having the correct field darkened, usually the "Subject:" field.)[19] Then push the **Include** button.

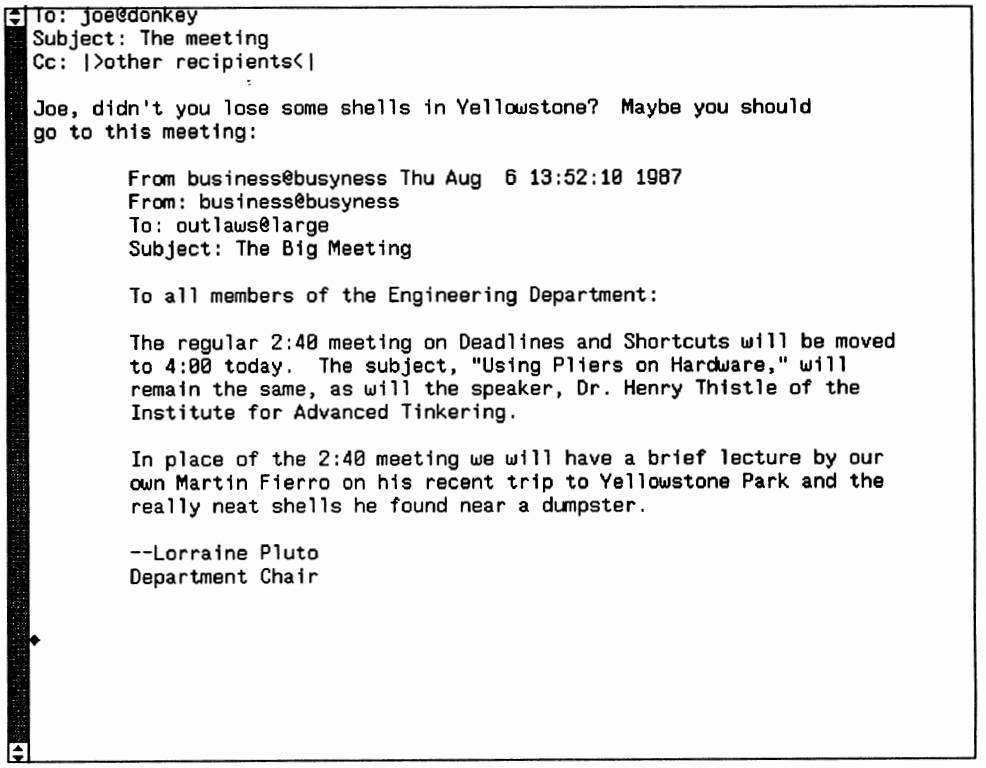
[19] See the *SunView 1 Beginner's Guide* for information on "insertion points."

Figure 2-21 *Including a Letter*



Including can also be done indented, by choosing the 'Include, Indented' option of the **Include** menu. Here's Figure 2-21 done with an indented include:

Figure 2-22 *Including a Letter, Indented*

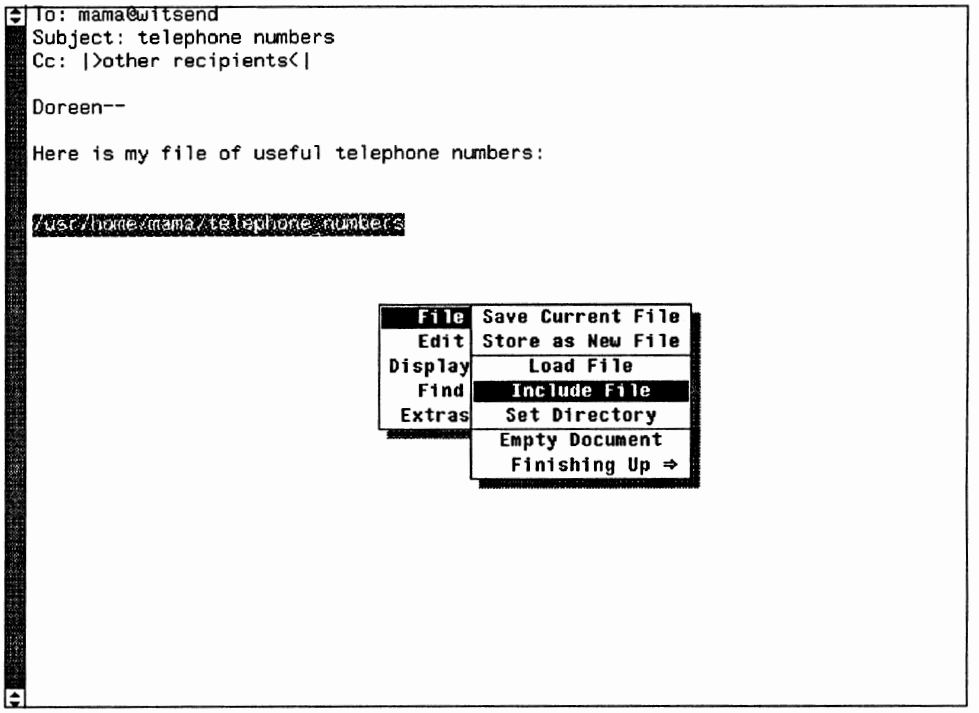


The included letter goes in at whatever place the caret is. (See the *SunView 1 Beginner's Guide* for more on "insertion points.")

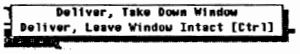
Compose and **Reply**, in the Mail Tool panel window, have menus with 'Include' on them. So you can include a letter directly when you bring up a composition window, instead of using the **Include** button.

You can also include any old file in a letter by using the Text Edit menu that all SunView text windows have. The 'Include File' menu item will insert a file into the composition window. See the *SunView 1 Beginner's Guide* for more information on the Text Edit menu.

Figure 2-23 *The Text Edit Menu in Mail Tool*



The Deliver Button



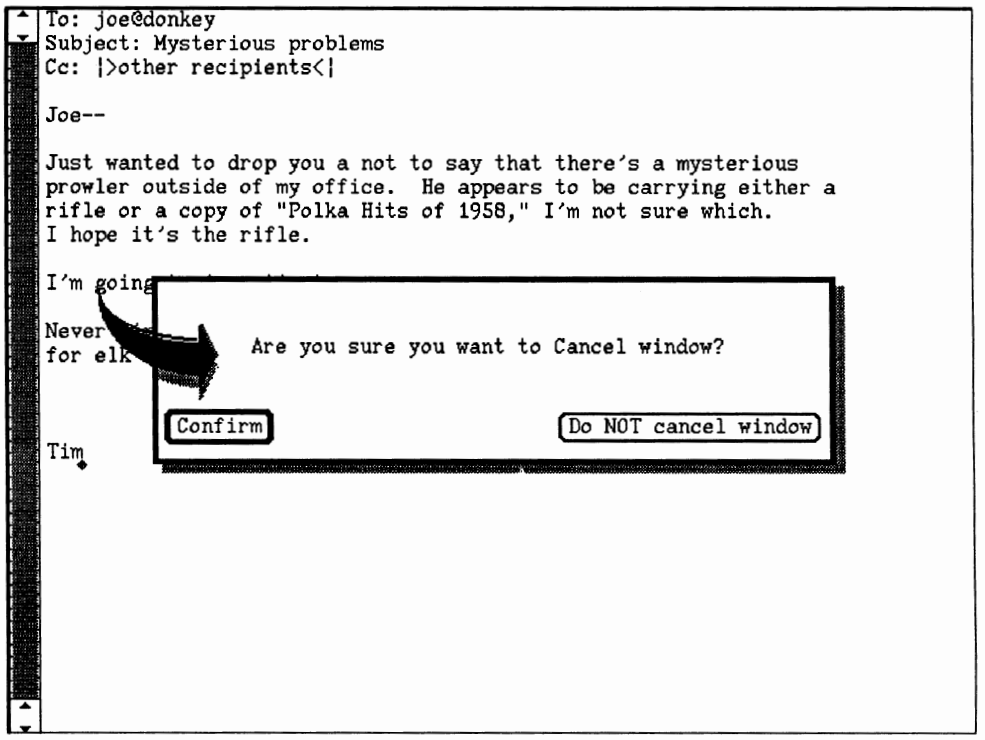
Once you've written your letter, you send it with the **Deliver** button. You should be sure that your letter's in the form you want it to be when you push **Deliver**; once a letter is sent, you can't take it back!

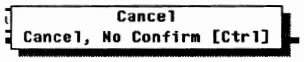
The **Deliver** menu allows you to send a letter but leave the composition window intact; ie., with the letter still in it. This is useful if you want to send the letter off to someone else (possibly editing it first), or to keep a copy for yourself.

The Cancel Button

You can, however, cancel a letter *before* you send it. If you press the **Cancel** button, Mail Tool will ask you to confirm the cancellation.

Figure 2-24 *Confirming a Cancellation*

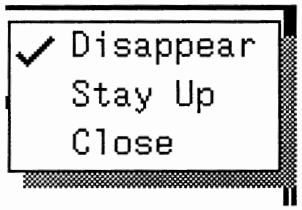




You can avoid this confirmation query by choosing 'Cancel, No Confirm' from the **Cancel** menu or by holding down the Ctrl key when you push the **Cancel** button.

You can modify Mail Tool so that it will not ask you to confirm certain irreversible operations such as quitting windows or cancelling a composition. You do this by turning the `expert` variable on. This is explained in Appendix A.

The Disappear Cycle Button



The **Disappear** cycle button affects the behavior of the composition window when you push **Cancel** or **Deliver**. Normally, when you deliver or cancel a letter, the composition window folds up and goes away. If you know you're going to want to write another letter, though, you can set the composition window to stay up after you deliver or cancel a letter. When the composition window is a pop-up window (ie., not split off from the message window), you can also set it to close itself to iconic form after cancelling or delivering.[20] Mail Tool reuses a composition window closed in this fashion, instead of starting new ones.

[20] The window will stay up if you choose 'Leave Window Intact' from the **Deliver** menu, regardless of how the **Disappear** cycle is set.

Note that the **Cancel** button changes to **Clear** when the **Disappear** cycle is set to 'Stay Up'.

When the **Disappear** button is set to 'Stay Up' and you cancel or deliver a letter, the composition window goes blank. To put the address template back into the window, push the **Re-address** button.

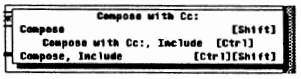
A Word about Word-Wrap

As noted in the *SunView 1 Beginner's Guide*, you can set an automatic word wrap in text editing windows, including Mail Tool composition windows. That means that SunView will automatically break lines you type between words, so you don't have to type a carriage return at the end of each line. However, using word wrap can cause difficulties when composing a letter. If the person you're sending the letter to doesn't also use word wrap — suppose he or she isn't using SunView, for instance — then the letter you send may come out with very long lines. These lines may run out of his or her message window, or the lines may be split in the middle of words instead of between them. For this reason we suggest that, when starting out, you do not use word wrap in Mail Tool.

Because line wrap is so convenient, however, we include in Appendix B an explanation of how you can get your Mail Tool (and `mail`) letter "prettied up" as you send them out. By following the set-up described there, you can use word wrap and know that your letters will come out legible to your readers.

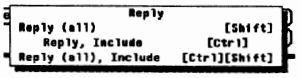
If you receive a ragged letter from someone, you can pretty it up by choosing 'Fmt' from the Text Edit menu available with Mail Tool. See the *SunView 1 Beginner's Guide* for more on Text Edit.

The Compose Button



Now that you've read about the `Cc:` option and including letters in compositions, the **Compose** button menu should be easy for you to figure out. It allows you to compose with or without a `Cc:`, and with or without including a received letter.

2.14. Replying to Mail



You reply to whichever letter you've selected from the header list, regardless of whether it's displayed in the message window.

The **Reply** button is very similar to the **Compose** button. They both bring up a composition window; the difference is that the **Reply** button automatically addresses your letter to whomever you're replying to.

Like the **Compose** button, **Reply** allows you to include the original letter as part of your reply; it has a menu with 'Include' on it.

The **Reply** button also allows for you to reply to everyone who was sent the original letter. You do so with the 'Reply (all)' or 'Reply (all), Include' menu items. That means that if you are one of ten people who received a letter, you can send your reply to everyone else who received it, as well as its author.

2.15. Closing and Exiting Mail Tool

You can leave Mail Tool up and running in its open form as long as you like. Sooner or later, though, you're probably going to want to close or exit it. In either case you still receive mail in your mailbox.

Closing Mail Tool

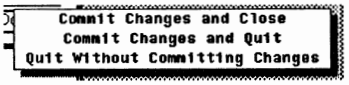
Closing Using SunView:

- You can close Mail Tool the way you close any other SunView window, as explained in the *SunView 1 Beginner's Guide*.(For example, you can use the [Open] key on the keyboard.) This is the fastest way to close Mail Tool.

If you use the mouse or an accelerator to close Mail Tool like a regular window, then you won't perform a commit. That means that any letters which you have deleted will not be permanently removed, and when you open Mail Tool again you'll still be able to undelete them. However, they'll still be in your mailbox, taking up room — which means that if you always open and close Mail Tool this way, you'll have a mailbox full of temporarily deleted letters filling up your file system. It's a good idea to do a commit now and then, just to keep your mailbox a manageable size.

Closing Mail Tool this way means that Mail Tool will not look for new mail automatically when you open it again. This saves time if you know that you won't want to look for mail, but it means that you must push the **New Mail** button if you do want to search for new mail.

Closing Using **Done**:



- The other way to close Mail Tool to its iconic form is by using the **Done** button. When you push **Done**, Mail Tool commits all changes before closing itself. When you next open Mail Tool, it will automatically look for new mail (see Section 2.2).

Quitting Mail Tool

Exiting using SunView:

- You can quit Mail Tool as you would quit any SunView window, from its frame menu, as explained in the *SunView 1 Beginner's Guide*. If you do so, Mail Tool will perform a commit before it disappears, so any changes will be reflected the next time you start Mail Tool.

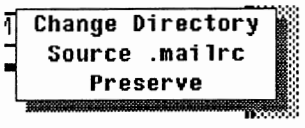
Exiting using **Done**:

- You can also use the **Done** button to quit. There are two ways to do this:

  If you choose 'Commit Changes and Quit' from the **Done** menu, Mail Tool will commit all your changes before disappearing.

  Choosing 'Quit Without Committing Changes' also exits Mail Tool. When you start a new Mail Tool, you'll find that any letters you've deleted without committing will still be there, as though you'd never deleted them.

2.16. The Misc Button



There is one last button we haven't talked about yet: **Misc**, for *miscellaneous*. **Misc** has three menu choices.

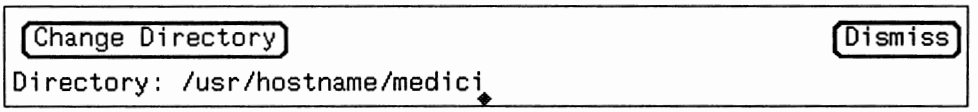
First, you can change the directory that Mail Tool is active in with the 'Change Directory' menu option. This means that Mail Tool will use that directory for finding and saving files. Remember that if you type

rug_fiction

in after `File:` Mail Tool looks for `rug_fiction` in its current directory. (See Section 2.10 — Files and Folders). 'Change Directory' simply changes the current directory. By using 'Change Directory' you save and retrieve `rug_fiction` in a different place. (It does not affect where Mail Tool looks for folders, however; the folder directory stays the same.)

You can also use 'Change Directory' to find out what Mail Tool's current directory is. When you choose 'Change Directory', you get a pop-up window which looks like Figure 2-25. This pop-up window gives you the current directory and asks you for the name of a new directory to switch to.

Figure 2-25 *The Change Directory Pop-up Window*



Second, the 'Source .mailrc' option modifies Mail Tool's behavior. Recall that `.mailrc` is the file which contains settings affecting Mail Tool. (This is covered in Appendix A, Modifying Mail and Mail Tool). Normally when you modify `.mailrc` you have to bring up a new Mail Tool to see your changes take place. Choosing 'Source .mailrc' allows you to incorporate changes to `.mailrc` directly into the Mail Tool you're currently running.

There is a limitation on 'Source .mailrc', however: If you change the setting of a variable from not set to set (ie., from turned off to turned on), then 'Source .mailrc' causes the change to take effect. If you change the variable from being set to not set (on to off), then it won't; i.e., 'Source .mailrc' doesn't cause Mail Tool to forget what options you've previously set. To get Mail Tool to work with such a changed `.mailrc`, you have to quit Mail Tool and start a new one after changing `.mailrc`.

Finally, **Misc** includes the 'Preserve' menu item. We briefly noted in Section 2.5 that you can turn off the `hold` variable in `.mailrc`, meaning that letters you read will not be held in your mailbox for saving or deletion, but will be transferred to a file or folder you designate.[21] (The transfer occurs when you commit changes.) The 'Preserve' option overrides this transfer on a letter-by-letter basis: a preserved letter is kept in your mailbox after a commit. Since the `hold` variable is normally turned on in Mail Tool, 'Preserve' usually has no effect and is not needed. (Note that `hold` is turned *off* for `mail`; See Appendix A for more information.)

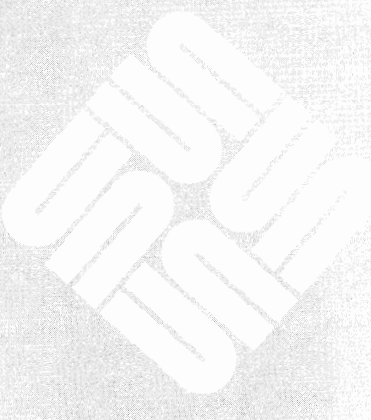
2.17. Conclusion

This concludes your introduction to Mail Tool. You may want to read the chapter on `mail` for more information on sending electronic mail on SunOS and UNIX.

[21] Usually a file called `mbox` in your home directory.

3

Mail



3

Mail

3.1. Sending Mail

The easiest way to use `mail` is to send a letter to someone, even yourself. In fact, start by sending a letter to yourself, so you can make sure that you have the hang of it before subjecting someone else to your trials, and so that you will have a letter you can read "in your mailbox."

To send a letter, type `mail` followed by the *username* of the recipient.

Sending a Letter To Yourself

To send a letter to yourself, type `mail` followed by your *username*. Then, type the text of the letter, on as many lines as you wish. Type your end-of-file character, usually Ctrl-D, on a line by itself to terminate input of the letter text.[22]

Figure 3-1 *Sending a Letter To Yourself*

```
venus% mail medici
Subject: Mail To Myself
Introspection is a narcissistic pursuit.
D
venus%
```

`mail` changes the `^D` on the screen to `EOT` to confirm the end of text of your letter.

Sending a Letter to Someone Else

Section 1.2 explains how electronic mail addressing works. Briefly, to send mail to someone on the same machine as yourself, give just his or her username as the destination. To send mail to people on other machines, give the username, followed by the @ symbol and the name of that person's machine. However, if your facility supports the Yellow Pages alias map, you should be able to send mail to users on other machines just by giving the person's username.

(If your machine isn't on a network, you can't send mail to users on other machines.)

Chapter 6 provides information on sending letters over remote networks.

---

[22] In these examples, `venus%` is the command prompt because "venus" is the name of your example machine. `medici` is your example username.

Here's how `medici` would send a letter to user `watson`. In this case we assume that `watson` is either on the same machine as `medici`, or that their network supports the Yellow Pages alias mapping:

Figure 3-2 *Sending a Letter to Someone Else: I*

Note: Username `watson` probably doesn't exist on your machine or local network. The next section describes the result of sending mail to a nonexistent username.

```
venus% mail watson
Subject: Tentative communication
Come here Watson.
I need you.    (Type Ctrl-D on next line to end text, send letter.)
EOT
venus%
```

(Sometimes you'll get a prompt when you type Ctrl-D. See under `askcc` in Appendix A and Section 3.14.)

If `medici` were not on a Yellow Pages system and wanted to send a letter to user `muddy` on the machine `waters`, this is what he would type:

Figure 3-3 *Sending a Letter to Someone Else: II*

When in doubt, include both the username and the machine.

```
venus% mail muddy@waters
Subject: Repair Job
My mojo's not working.
Can you help me fix it?    (Type Ctrl-D)
EOT
venus%
```

It takes a little while for the mail facility to deliver mail.

To specify multiple recipients for your letter, type more than one username, each separated by a space character.

Sending Mail to a Nonexistent Username

If you send the letter to a username that does not exist, `mail` will realize, after a minute or two, that it cannot deliver the letter.

For example, suppose you send mail to nonexistent username `amorphous`, like this:

Figure 3-4 *Sending a Letter to a Nonexistent Username*

```
venus% mail amorphous
Subject: Greetings!
What are you up to,
old pal?    (Type Ctrl-D on next line to end text, send letter.)
EOT
venus%
```

At this point, three things happen:

1) The mail facility displays an error notification containing the nonexistent username, followed by three dots (`. . .`) and `User unknown`. In the case of the above example, the error message is:

```
amorphous... User unknown
```

2) The mail facility delivers a letter to you, the originator of the faulty letter that looks something like this:

Figure 3-5 *Mail Facility Message To Originator When User Unknown*

```
From: medici Thu Oct 31 23:59:59 1985
Return-Path: <MAILER-DAEMON>
Received: by venus.sun.com (4.0/SMI-4.0)
    id AB09802; Thu, 31 Oct 85 23:58:59 PDT
Date: Thu, 31 Oct 85 23:58:59 PDT
From: MAILER-DAEMON (Mail Delivery Subsystem)
Subject: Returned mail: User unknown
Message-Id: <8510220038.AB09802@venus.sun.com>
To: medici

   ----- Transcript of session follows -----
550 amorphous... User unknown

   ----- Unsent message follows -----
Return-Path: <medici>
Received: by venus.sun.com (4.0/SMI-4.0)
    id AA09798; Thu, 31 Oct 85 23:58:59 PDT
Date: Thu, 31 Oct 85 23:58:59 PDT
From: medici (Cosimo de' Medici)
Message-Id: <8510220038.AA09798@venus.sun.com>
To: amorphous
Subject: Greetings!

What are you up to,
old pal?
```

3) The mail facility delivers a letter that looks something like this to your machine's Postmaster:

Figure 3-6 *Mail Facility Letter Postmaster When User Unknown*

```
From: medici Thu Oct 31 23:59:59 1985
Return-Path: <MAILER-DAEMON>
Received: by venus.sun.com (4.0/SMI-4.0)
    id AA09802; Thu, 31 Oct 85 23:58:59 PDT
Date: Thu, 31 Oct 85 23:58:59 PDT
From: MAILER-DAEMON (Mail Delivery Subsystem)
Subject: Returned mail: Mail problem
Message-Id: <8510220038.AA09802@venus.sun.com>
To: Postmaster

   ----- Transcript of session follows -----
550 amorphous... User unknown

  ----- Message header follows -----
Return-Path: <medici>
Received: by venus.sun.com (4.0/SMI-4.0)
    id AA09798; Thu, 31 Oct 85 23:58:59 PDT
Date: Thu, 31 Oct 85 23:58:59 PDT
From: medici (Cosimo de' Medici)
Message-Id: <8510220038.AA09798@venus.sun.com>
To: amorphous
Subject: Greetings!
```

Note: The *Postmaster* for a given machine is a username designated to receive notice of letters that the mail facility cannot deliver. Either you or your system administrator will probably be the Postmaster for your machine.

The mail facility delivers to the Postmaster notice of the letter you sent *without* delivering the letter's text; so your mail is still somewhat confidential, even when you make a mistake.

Aborting a Letter

When you have started to send a letter, but you decide you no longer want to send it, type your *interrupt character*, usually Ctrl-C, to abort the letter. `mail` displays a message asking you to confirm the letter abort by typing Ctrl-C once again. `mail` won't send a letter when you abort it using the second Ctrl-C.

Figure 3-7 *Aborting an Attempt at Sending a Letter*

```
venus% mail nowhere
Subject: Over the Rainbow
Some electronic mail is not meant for anyone to^C
(Interrupt -- one more to kill letter)
^Cvenus%
```

Note: When you want to abort a letter while typing the subject, you must type Return after the first Ctrl-C to get `mail` to interpret the interruption properly.

3.2. Reading Mail

To start `mail` so that you can read your letters, type `mail`, without any arguments, to the command prompt.

Figure 3-8 *Starting* `mail` *To Read Letters*

```
venus% mail
Mail version SMI 4.0 Mon Feb 17 00:20:58 PDT 1986  Type ? for help.
"/usr/spool/mail/medici": 2 messages 2 new
>N  1 medici              Thu Oct 31 23:59  12/323   Mail To Myself
 N  2 MAILER_DAEMON       Fri Nov  1 00:02  26/725   Returned Mail: User un
&
```

`mail` displays the program name, program version number, and version date, then informs you that you can type a question mark (`?`) to get help information.

On the second line, `mail` specifies which file it picks up your mail from, in other words your *mailbox*, tells you how many letters you have, and whether they are *new* or *unread*. In the example above, the mailbox directory is `/usr/spool/mail/medici`, with two new letters.[23]

Note: All letters that are neither new (`N`) nor unread (`U`) are old letters that you have read (no status indicator). `mail` marks letters you've saved to a file or folder with an asterisk character (`*`).

Starting on the third line, `mail` displays a numbered list of the letters in your mailbox. Each of these lines specifies:

letter status	New (`N`), unread (`U`), or old (no letter status listed)
letter number	Number you can use to specify that letter
sender	Name of user (sometimes machine) letter came from
time sent	Date and time sender sent the letter
size	Number of lines, number of characters, in letter
subject	Subject of the letter

Note: The *current letter* is the letter that you last read, or the first letter you read by default when obtaining new mail. Within the numbered list of letters, a greater-than symbol (`>`) prefaces the current letter listing.

The line beginning with a greater-than symbol (`>`) is the *current letter*.

In the example, the `N` means the letter is new; `1` is the letter number; `medici` is the sender; `Thu Oct 31 23:59` is the date and time `medici` sent the letter, and `12/323` means there are 12 lines and 323 characters in the letter.

Finally, `mail` displays an ampersand prompt (`&`) to let you know you can type `mail` commands.[24]

If you start `mail` when you don't have any letters waiting for you in your mailbox, you will see something like this:

[23] Actually, the mailbox is located in `/var/spool/mail`, but SunOS maintains a symbolic link from `/usr` for compatibility with older mail programs.

[24] You can change this prompt to something else if you like. See `prompt` in Appendix A.

Figure 3-9 *Starting* `mail` *with an Empty Mailbox*

```
venus% mail
No mail for medici
venus%
```

Each time you log in, your machine informs you if you have mail in your system mailbox by displaying

```
You have mail.
```

on a line by itself just after your regular login letters.

3.3. How To Read Letters

Once you have entered `mail` and examined the numbered list of letters, you can read a given letter by typing its *letter number* to the `mail` prompt.

For the initial example above, there are two letters, with letter numbers one and two. Type `1` to the `mail` prompt, and `mail` displays the first letter.

Figure 3-10 *Reading a Letter*

```
venus% mail
Mail version SMI 4.0 Mon Feb 17 00:20:58 PDT 1986  Type ? for help.
"/usr/spool/mail/medici": 2 messages 2 new
>N  1 medici             Thu Oct 31 23:59  12/323   Mail To Myself
 N  2 MAILER_DAEMON      Fri Nov  1 00:02  26/725   Returned Mail: User un
& 1
Message  1:
From medici Thu Oct 31 23:58:59 1985
Return-Path: <medici>
Received: by venus.sun.com (4.0/SMI-4.0)
        id AA12623; Thu, 31 Oct 85 23:59:59 PDT
Date: Thu, 31 Oct 85 23:59:59 PDT
From: medici (Cosimo de' Medici)
Message-Id: <8510232235.AA12623@venus.sun.com>
To: medici
Subject: Mail To Myself
Status: R

Introspection is a narcissistic pursuit.

&
```

3.4. Looking at the Numbered Letter List

Now that you have read the first letter in your mailbox, when you next look at the numbered letter list, the `N` status of that letter no longer appears.

To look at the numbered letter list, type `headers`, or just `h`, to the `mail` prompt.

Figure 3-11 *Looking at the Numbered Letter List*

```
. . .

& h
>   1 medici            Thu Oct 31 23:59  12/323   Mail To Myself
 N  2 MAILER_DAEMON     Fri Nov  1 00:02  26/725   Returned Mail: User un
&
```

The `N`, for new letter, no longer appears just after the greater-than sign; you changed the status of the letter when you read it.

3.5. Reading the Current Letter

Instead of specifying the letter number, you could type `print` to the `mail` prompt to read the *current letter* in your mailbox. `p` works as an abbreviation for `print`.

Figure 3-12 *Reading the Current Letter*

```
. . .

& p
Message  1:
From medici Thu Oct 31 23:58:59 1985
Return-Path: <medici>
Received: by venus.sun.com (4.0/SMI-4.0)
        id AA12623; Thu, 31 Oct 85 23:59:59 PDT
Date: Thu, 31 Oct 85 23:59:59 PDT
From: medici (Cosimo de' Medici)
Message-Id: <8510232235.AA12623@venus.sun.com>
To: medici
Status: R

Introspection is a narcissistic pursuit.

&
```

As another alternative, when you want to read the next letter in the list, simply type Return.

3.6. Letter Format

What is all that stuff in the letter?

A letter has two parts: the *header* and the *body*. When you send a letter the quick way you learned in Section 3.1 , you don't see the header, except for the `Subject:` line. You'll see more of the header in Section 3.13, when you compose a letter. But mostly, you see the header when reading your mail.

The header comprises a number of lines. Each line describes attributes of the letter, such as:

`Return-Path:` Address used to return mail that is undeliverable

`Received:`	Machine, letter identification information, and arrival time for each machine along the letter's *network path.*
`Date:`	Time sent, including date
`From:`	Username (sometimes machine name) of letter sender
`Message-Id:`	Letter identification information
`To:`	Username (sometimes machine name) of letter recipient
`Subject:`	Subject of letter

You can set up `mail` so that it doesn't display unwanted letter header lines. See `ignore` in Section A.3 of Appendix A for more information. From now on, we will only show part of the header, ignoring the other lines.

3.7. Saving Letters in Files

To save a letter into a file, type `save`, or the abbreviation `s`, followed by the *letter number* and the *filename* of the file you want to contain the letter. `mail` responds by displaying the filename, followed by the status of the file, and the size of the file.

For example, to save the first letter in the example user's mailbox into the file `first.mail`:

Figure 3-13 *Saving a Letter into a File*

```
. . .

& h
>   1 medici              Thu Oct 31 23:59  12/323   Mail To Myself
 N  2 MAILER-DAEMON       Fri Nov  1 00:02  26/725   Returned Mail: User un
& s 1 first.mail
"first.mail" [New file] 12/333
&
```

Note: When you save a letter to a file, but you don't specify which letter number, `mail` assumes that you want to save the current letter into the filename you specify.

Now, the file `first.mail` contains the header and body of letter number one.[25]

3.8. Saving Letters in Folders

A *folder* is a mail file like any other, except that it is located in a special directory called a *folder directory*. You save, retrieve, delete, and copy letters in folders just as you would any mail file, but folders are often more convenient to use, for two reasons. First, when you store mail into folders, you know where the folder will be — in your folder directory. You don't have to worry about having mail files scattered all about your file system. Second, folders provide you with an easy shorthand: you can use a + (plus sign) to represent your folder directory;

[25] Unless you specify an absolute pathname for the file, it will be placed in the current directory. For information on "absolute pathnames," see *Getting Started with SunOS: Beginner's Guide.*

the + is an *abbreviation* for that directory's name. Here's an example: suppose you designate `/home/medici/mail` as your folder directory. Then the folder `+testmail` is interpreted by `mail` as `/home/medici/mail/testmail`. When you want to do something with that file, you type in only instead`+testmail`,

Before using folders, you must choose a directory with name *directory-name*, into which `mail` will locate all folders with names that have an initial plus sign. You do so by setting the `folder` variable in your `.mailrc` file to *directory-name*. This is explained in Appendix A.

To save a letter in a folder, type

`save` *letter-number* *+foldername*

to the `mail` prompt.

To save example letter number one into the folder `testmail`:

Figure 3-14 *Saving a Letter into a Folder*

```
. . .

& h
>*   1 medici              Thu Oct 31 23:59  12/323   Mail To Myself
 N   2 MAILER_DAEMON       Fri Nov  1 00:02  26/725   Returned Mail: User un
& save 1 +testmail
"+testmail" [New file] 12/333
&
```

Note: The asterisk (`*`) in the numbered letter list indicates that you have saved that letter into a file or folder. The asterisk replaces any new (`N`) or unread (`U`) letter status indicator when you save a letter.

For more on folders, see section 2.10.

3.9. Quitting `mail`

To quit `mail`, type `quit`, or the abbreviation `q`, to the `mail` prompt.

The `quit` command moves any already-read messages you haven't saved in a file or folder from your mailbox into a file called `mbox` in your home directory.[27] For example, when you have two messages which you read, but didn't save in a file or folder, `mail` displays the notification:

```
Saved 2 messages in /home/venus/medici/mbox
```

When you quit `mail` without reading a letter that appeared in the numbered letter list, `mail` will hold the unread letter in your mailbox, marked as unread.

[26] The + is not part of the filename, and is invisible outside of `mail`.

[27] This is different from Mail Tool, which does not use the `mbox` scheme. See the `hold` and `MBOX` variables, described in Appendix A.

Figure 3-15 *Quitting* `mail`

```
. . .

& q
Saved 2 messages in /home/venus/medici/mbox
Held 1 message in /usr/spool/mail/medici
venus%
```

3.10. Reading Letters in a File

You can use an editor to look at letters that you've saved in a file, or you can read the letters with the `mail` program. To use `mail`, type `mail` followed by the option `-f` *filename*. For instance, to read the example letter saved in the file `first.mail`:

Figure 3-16 *Reading a Letter Saved in a File*

```
venus% mail -f first.mail
Mail version SMI 4.0 Mon Feb 17 00:20:58 PDT 1986  Type ? for help.
"first.mail": 1 message 1 new
>   1 medici            Thu Oct 31 23:59  12/323   Mail To Myself
& p
Message 1:
From medici Thu Oct 31 23:58:59 1985
From: medici (Cosimo de' Medici)
To: medici
Subject: Mail To Myself

Introspection is a narcissistic pursuit.

& q
"first.mail" complete
venus%
```

When you save a letter in a file, `mail` automatically removes it from your mailbox.[28] However, `mail` notifies you that the letter is in the file by displaying "*filename*" `complete`, or in the case of this example `"first.mail" complete`.

3.11. Reading Letters in a Folder

Reading a letter saved in a folder is similar to reading a letter saved in a file — type `mail` followed by the option `-f +`*foldername*. For instance, to read the example letter saved in the folder `+testmail`:

[28] Unless you have `keepsave` turned on; see Appendix A.

Figure 3-17 *Reading a Letter Saved in a Folder*

```
venus% mail -f +testmail
Mail version SMI 4.0 Mon Feb 17 00:20:58 PDT 1986  Type ? for help.
"+testmail": 1 message 1 new
>   1 medici            Thu Oct 31 23:59  12/323   Mail To Myself
& p
Message 1:
From medici Thu Oct 31 23:58:59 1985
From: medici (Cosimo de' Medici)
To: medici
Subject: Mail To Myself

Introspection is a narcissistic pursuit.

& q
"+testmail" complete
venus%
```

When you save a letter in a folder, `mail` removes it from your mailbox.[29] However, `mail` notifies you that the letter is in the folder by displaying *"foldername"* `complete`, or in the case of this example `"+testmail" complete`.

3.12. Deleting Letters

To delete a letter, type `d`, for delete, followed by a space character, and the *letter number* of the letter.

Figure 3-18 *Deleting a Letter*

```
venus% mail
Mail version SMI 4.0 Mon Feb 17 00:20:58 PDT 1986  Type ? for help.
"/usr/spool/mail/medici": 2 messages 2 new
>*  1 medici            Thu Oct 31 23:59  12/323   Mail To Myself
 N  2 MAILER_DAEMON     Fri Nov  1 00:02  26/725   Returned Mail: User un
& d 2
& h
>*  1 medici            Thu Oct 31 23:59  12/323   Mail To Myself
& q
venus%
```

3.13. Composing Letter

When you want to send a letter without worrying about mistakes you type in the letter's text, you can *compose* the letter using the `vi` text editor.[30] This section describes how to:

- Compose a letter using `vi` within the `mail` program
- Carbon copy the letter to other users
- Abort a letter

[29] Unless you have `keepsave` set. See Appendix A.

[30] For more information about `vi`, see the chapter on editing files in *Getting Started with SunOS: Beginner's Guide.*

- Compose a letter while reading your mail
- Send mail over local networks.[31]

Composing a Letter Using `vi`

To compose a letter using `vi`, first type a letter as you learned in Section 3.1. In other words:

- Type `mail`
- Wait for the `Subject:` prompt
- Type the subject of the letter
- Type Return
- Type any of the letter body text you desire

Next start `vi`:

- Type `~v` at the beginning of a line[32]
- Type Return.

The tilde character (`~`) signals `mail` to interpret the following character (`v`) as a command, in this case a command to start up the visual text editor `vi`.

Figure 3-19 *Starting* `vi` *From Within* `mail`

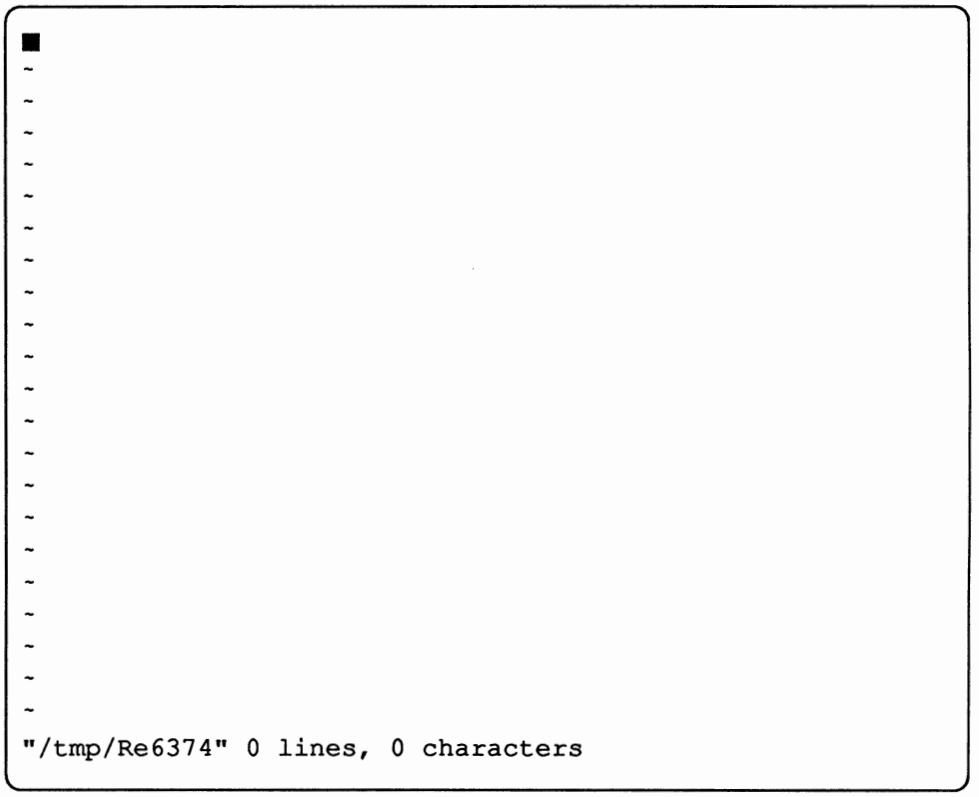
```
venus% mail wilde
Subject: The Importance of Being Earnest
~v
```

After a moment, the `vi` interactive screen appears, ready for you to edit an empty file located in your `/tmp` directory.

[31] For information on sending mail over remote networks, see Chapter 6 .

[32] Often the tilde (`~`) won't appear on the screen.

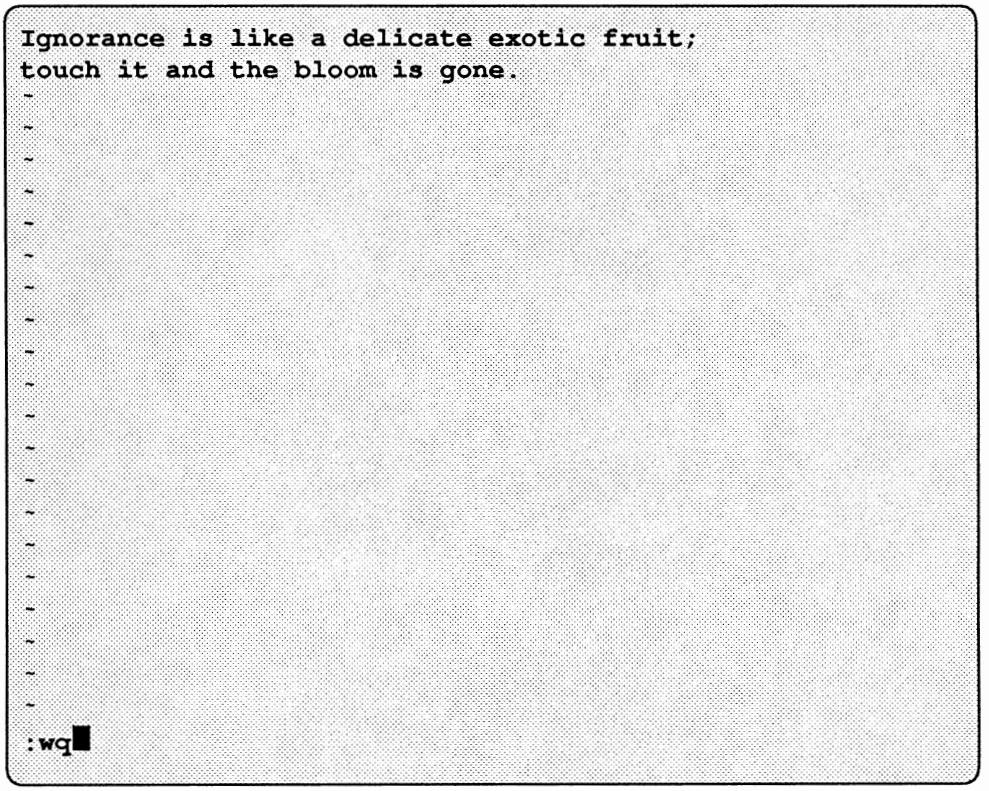
Figure 3-20 *The* `vi` *Interactive Screen and Temporary File*



Next, type `i` to insert the letter text.[33] Type the letter text, followed by ESC to tell `vi` that you've entered all of the text. Then, type a colon (`:`) to enter colon mode, followed by `wq` and a carriage return to write, or save, the file, and to quit `vi`.

[33] The text is from Oscar Wilde's *The Importance of Being Earnest.*

Figure 3-21 *Entering the Letter Text Using* `vi`



`mail` displays `(continue)` to let you know when it takes over from `vi`. At this point, you can type in more letter text, or type Ctrl-D to end the letter text and send the letter. Once again, the `^D` symbol on the screen changes to `EOT`, end of text, as `mail` acknowledges your instructions and sends the letter.

Figure 3-22 *Returning from* `vi` *to* `mail` *and Sending the Letter*

```
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/tmp/Re6374" 2 lines, 75 characters
(continue)
EOT
venus%
```

You have successfully composed and sent a letter using `vi` within `mail`.

Getting A List of Commands

When you can't remember one of the `mail` commands beginning with the tilde character, type the tilde character, followed by a question mark (`~?`) and Return, at the beginning of a line to get a list of such commands.

3.14. Adding a Carbon Copy to Someone Else

Before you send your letter, you can specify that a *carbon copy* of the letter go to additional users.

Do so by typing the tilde character (`~`), letter `c`, a space character, and the usernames you want to receive a carbon copy, each separated by a space character. Then send the letter as usual by typing Ctrl-D.

You can modify `mail` so that it always prompts you with `Cc:` at the appropriate point. See Appendix A for more information on the `askcc` variable.

3.15. Confirming a Letter

Before you send a letter, you can look it over to see if you want to change it at all. You do so, while you're still composing it, with the `~p` and `~h` commands.

The `~p` command displays the contents of the letter (without its header). It then returns you to the point that you were at when you typed `~p`, and then it displays

```
(continue)
```

to let you know that the display is finished and that you're back in insertion mode. You can then use `vi` to make any changes you want.

The `~h` command is similar to the `~p` command except that it displays the header — ie., the **To:** and **Subject:** lines. Unlike the `~p` command, however, it doesn't display the header all at once; instead, it displays each line individually, ready for editing. If you've already typed a subject or destination in, it's displayed so that you can change it if you so choose.

The `~h` command automatically prompts you with `Cc:` as well. (See section 3.14.) It also prompts you with `Bcc:`, which stands for *blind carbon copy*. Like `Cc`, `Bcc` copies the letter to the addressee you specify; unlike `Cc`, however, the addressee's name doesn't show up in the letter. You use `Bcc` when you want to secretly send someone a letter: if you send Mary a letter and `Bcc` Tom, both Mary and Tom will receive it, but only Mary's name will appear as a recipient.

3.16. Aborting a Letter

Aborting a letter from within `vi` isn't all that different from aborting a regular letter. Quit `vi` to return to `mail` — type

ESC :wq Return

then, when you're back in `mail`, type Ctrl-C twice in succession.

3.17. Composing a Letter While Reading Your Mail

When you want to compose a letter while reading your mail, starting with the ampersand `mail` prompt (`&`), rather than the command prompt, you can type `m` followed by the *username(s)* of the mail recipient(s) and Return.

`mail` responds just like it does when you compose a letter starting from the command prompt. So type:

- The letter subject to the `Subject:` prompt
- Return
- The letter text
- Ctrl-D to send the letter

You'll end up back at the ampersand mail prompt where you can continue to read your mail.

3.18. Replying to Mail

After you read a letter, you may want to *reply* to it, to answer questions or make comments. When you reply to the letter, you can insert a copy of the letter you're answering into your reply letter.

Reply by typing `reply` *letter number* to the `mail` prompt. `r` works as an abbreviation for `reply`.

The example shows how you can read a letter, then reply by typing only `r`, for reply. `mail` assumes you're replying to the current letter when you don't

specify a letter number.

`mail` constructs the `To:` and `Subject:` lines automatically from the letter you're answering. It replies to the sender of the original letter and precedes a copy of the original subject line with the string `Re:`.

Type Ctrl-D on a line by itself to send the mail when you've finished entering the letter text.

Figure 3-23 *Replying to a Letter*

Note: For the purpose of the examples in this section, assume that users `sappho` and `rimbaud` sent the example mail letters. You can generate mail by sending mail to yourself or waiting for a colleague to send you some.

```
venus% mail
Mail version SMI 4.0 Mon Feb 17 00:20:58 PDT 1986  Type ? for help.
"/usr/spool/mail/medici": 2 messages 2 new
>N  1 sappho@aphrodite   Thu Oct 31 23:59  21/391   Love and Sun
 N  2 rimbaud@verlaine   Fri Nov  1 00:02  16/515   Vagabonds
& p
Message  1:
From sappho@aphrodite Fri Nov  8 13:09:46 1985
From: sappho@aphrodite (Sappho)
To: medici@venus
Subject: Love and Sun

I confess

I love that
which carresses
me. I believe

Love has his
share in the
Sun's brilliance
and virtue

& r
To: sappho@aphrodite
Subject: Re:  Love and Sun

In the words of the popular band Black Flag:
"Who needs love when you've got a gun?"
EOT
&
```

As usual, `mail` confirms the end of text of the letter with `EOT`, then sends the letter.[34]

3.19. Inserting a Copy of a Letter

To insert a copy of the letter to which you are replying within the text of the reply:

Note: Inserting a copy of a letter with `~m` *letter number* is similar to entering `~v` to get into `vi` from `mail`.

- Reply to the letter using `r` *letter number*
- Typing the tilde character (`~`) and `m`, for letter, and an optional *letter number*

[34] The poem is by Sappho.

- Type [Return].

Even though you can't see the text of the inserted letters, `mail` inserts the letter you specify into the letter you are preparing to send. `mail` confirms the operation by displaying the notification `Interpolating:` followed by the letter number, and `(continue)` on the next line.

The inserted letter appears indented eight characters from the left margin of the letter text.[35] This is useful when you want to further edit the letter with `vi`, adding pertinent comments right near the appropriate parts of the original letter.

End the letter text as usual by typing a [Ctrl-D] on a line by itself.

Figure 3-24 *Inserting a Letter into Your Reply*

```
. . .
& r
To: sappho@aphrodite
Subject: Re:  Love and Sun

~m
Interpolating: 1
(continue)
What a beautiful poem, my dear!
EOT
&
```

In example 3-25, user `sappho` receives a letter that looks like this:

[35] When you want to insert a letter without indentation, use `~f` instead of `~m`.

Figure 3-25 *Inserted Letter After Delivery of Reply*

In the example, user `sappho`'s machine name is `aphrodite`.

```
aphrodite% mail
Mail version SMI 4.0 Mon Feb 17 00:20:58 PDT 1986  Type ? for help.
"/usr/spool/mail/sappho": 2 messages 2 new
>N  1 medici              Fri Nov  8 14:13   13/374   Re:  Love and Sun
 N  2 medici              Fri Nov  8 14:14   33/722   Re:  Love and Sun
& 2
Message 2:
From: medici@venus Fri Nov 8 14:14:04 1985
From: medici (Cosimo de' Medici)
To: sappho@aphrodite
Subject: Re:  Love and Sun

        From sappho@aphrodite Fri Nov  8 13:57:32 1985
        From: sappho@aphrodite (Sappho)
        To: medici
        Subject: Love and Sun

        I confess

        I love that
        which carresses
        me. I believe

        Love has his
        share in the
        Sun's brilliance
        and virtue

What a beautiful poem, my dear!
& q
Saved 1 message in /home/venus/medici/mbox
Held 1 message in /usr/spool/mail/sappho
aphrodite%
```

3.20. Inserting a Copy of a File

Note: When you're inserting a copy of a file from another directory, use the absolute pathname to specify the filename.

Inserting a copy of a file into a letter is like inserting a copy of a letter into another letter. Start to send a mail letter, or reply to a letter as before, only type the tilde character (`~`) and `r` *filename*, followed by Return. `mail` will insert the file called *filename* into your letter. Type Ctrl-D to end the letter text and send the letter. We continue with user `medici` as he struggles to form a more satisfactory reply to `sappho`:

Figure 3-26 *Inserting a File into a Letter*

```
. . .
& r
To: sappho@aphrodite
Subject: Re:  Love and Sun

~r blank.verse
"blank.verse" 0/0
EOT
Null message body; hope that's ok
& q
Held 1 message in /usr/spool/mail/medici
venus%
```

(Make sure you understand the difference between `~r` and `r`).

When you send a letter that doesn't contain any characters in its body, `mail` presents the notification

```
Null message body; hope that's ok
```

while sending the letter.

With this method of inserting a file into letters, you can insert files into an original letter as you compose it, or into letters to which you're replying.

3.21. Conclusion

For more information about `mail`, like how to reply to a letter with a copy to all the recipients of the original letter, see the `mail` Man Page, online or in the *SunOS Reference Manual.* Appendix A describes how you can change the behavior of `mail`.

4

Messages

4

Messages

This chapter describes messages, so that you can communicate with other users more immediately and interactively than by electronic mail.

There are three kinds of electronic messages:

- Interactive messages with `talk` or `write`
- Broadcast messages with `wall`
- System messages from your machine

4.1. Interactive Messages: `talk`

With the `talk` program, you can converse on your screen with someone else who is either using a terminal on your machine, or using another machine on your local network.

Note: When using the window system, initiate your `talk` message session in the window you want to use for the session. Pick a window that is large enough to contain a fair amount of text.

To start `talk`, type

`talk` *username@machine-name*

to your command prompt, followed by Return. In this example, user `medici` attempts to contact user `michaelangelo`.

Figure 4-1 *Starting a* `talk` *Message Session*

```
venus% talk michaelangelo@david
```

`talk`'s interactive screen appears and `talk` attempts to connect with the other user's machine. Until `talk` connects to the other machine, it displays the notification:

```
[No connection yet]
```

Once connected, `talk` notifies you that it is waiting for the other person to respond:

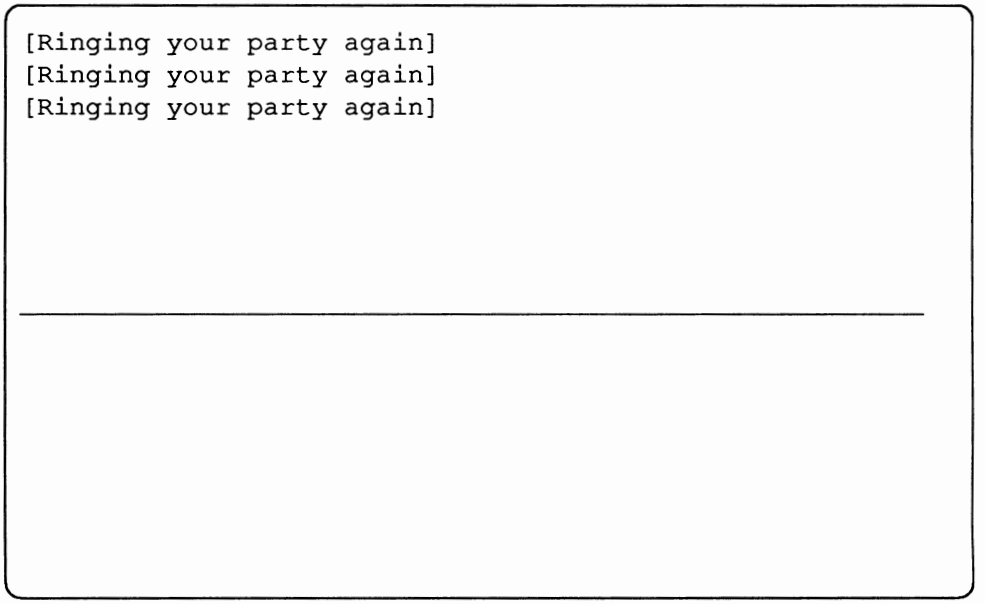
```
[Waiting for your party to respond]
```

`talk` "rings" the other person again and again, printing a message repeatedly on the screen while waiting for a response. If the other person isn't a user, or isn't logged at that time, `talk` responds with:

```
[Your party is not logged on]
```

But when `talk` finds the other user, the `talk` interactive screen displays a line to split itself in half like this:

Figure 4-2 `talk`*'s Interactive Screen*



To facilitate a connection, `talk` displays a message that includes your username and machine name on the other user's screen. In the case of example username `medici`'s attempt, `talk` displays the following message on user `michaelangelo`'s screen:

Figure 4-3 `talk` *Notifies the Other User*

```
Message from Talk_Daemon@venus at 0:01 ...
talk: connection requested by medici@venus
talk: respond with:  talk medici@venus
```

The other user must respond by typing `talk` followed by the *username* and *machine name* of the person who is attempting to talk. In our example, `michaelangelo` types:

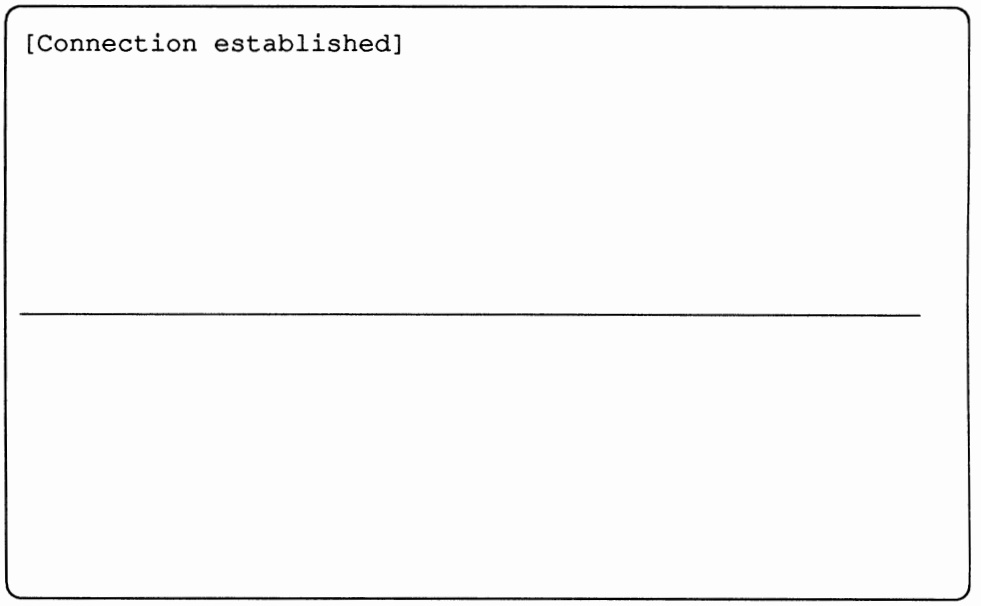
```
talk medici@venus
```

to confirm the `talk` connection with user `medici` on machine `venus`.

If `michaelangelo` is busy, or wants to ignore `medici`, he refuses to answer `medici`'s request, and eventually `medici` gives up, typing Ctrl-C to exit from the `talk` interactive screen.

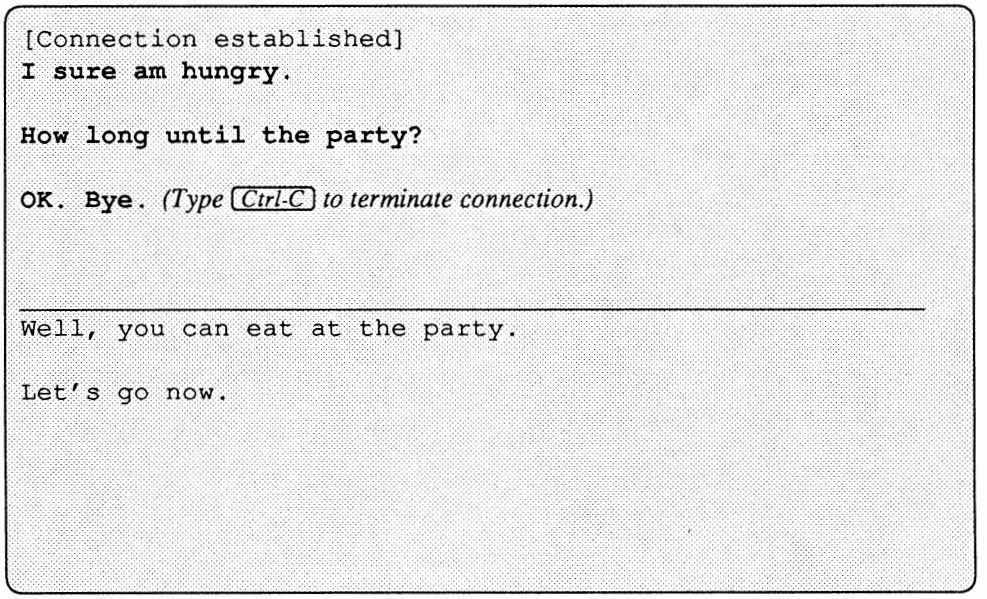
However, if `michaelangelo` successfully responds to `medici`'s request, `talk` establishes a link between the two users.

Figure 4-4 `talk` *Establishes Message Link Between Two Users*



Now, both users can type messages on the screen at the same time without interfering with each other. Both users see the messages they've typed on the upper half of their own screens or windows; the other user's messages appear on the lower half of their screens.

Figure 4-5 *Chatty* `talk` *Screen*



When they have finished typing messages, *either* user types Ctrl-C to terminate the `talk` message session.

As explained in Chapter 1 (Overview), you can prevent `talk` messages from appearing on your screen by typing

```
mesg n
```

If you're running the SunView window system, you must be superuser (`root`) to run `mesg`. You can also put `mesg n` on a line by itself in your `.login` file; this file is consulted whenever you log in, so messages will be turned off until you take the line out and log in again. *Doing More with SunOS: Beginner's Guide* contains information on becoming superuser, and *Setting Up Your SunOS Environment: Beginner's Guide* explains the `.login` file.

For more information on `talk`, see the `talk` Man Page, online or in the *SunOS Reference Manual.*

4.2. Interactive Messages: `write`

`write` differs from `talk` because `write`:

- doesn't use the entire screen or window
- only reaches users on the same machine or workstation you're sending messages from

One user `writes` a message to the other. Then, the other user can `write` a reply, reply in some other way, or decide not to reply.

To `write` a message to someone using a terminal on your machine, type `write` *username* to the command prompt, followed by Return.

In this example, user `medici` decides to `write` some messages to user `sappho`. He types in the text of an introductory message on lines following the `write` command line. To send the introductory message text, type Return.

Figure 4-6 *Writing a Message to Another User*

```
venus% write sappho
Do you want to
chat?    (Type Return to send message text.)
. . .
```

The message appears on the other user's screen almost immediately afterwards.

`sappho` decides to exchange messages with `medici`, so she types `write`, followed by his username, Return, and her message in reply.

Figure 4-7 `write` *Message Appears on Another User's Screen*

```
venus%
Message from venus!medici on ttyp2 at 1:01 ...
Do you want to
chat?
write medici
Sure, what's up?    (Type Return on next line to send message text.)
. . .
```

As you can see, `write` automatically identifies the machine, username, and terminal where the message originated, and the time the message arrived.

The two conversationalists can continue to write messages back and forth, without retyping the `write` command, until they want to stop. Then, *both* users must type Ctrl-D on a line by itself to terminate the `write` connection.

Figure 4-8 *Terminating a* `write` *Connection*

```
venus% write sappho
Do you want to
chat?
Message from venus!sappho on ttyp3 at 1:02 ...
Sure, what's up?
Oops, I'm late for an
appointment - gotta
run!    (Type Ctrl-D on next line to send message, terminate connection.)
venus%
```

`write` displays the end-of-file indicator, `EOF` on the other user's screen (for this example, user `sappho`'s screen) to notify that person that her conversational partner (user `medici`) terminated the connection.

Just as with `talk`, you can prevent `write` messages from appearing on your screen by using `mesg n`.

For more information on `write`, see the `write` Man Page, online or in the *SunOS Reference Manual.*

4.3. Broadcast Messages: `wall`

Note: Most users sharing a machine don't appreciate people who send spurious messages to everyone on the machine.

When you want to send a message to everyone on your machine at once, use the `wall`, write to **all**, command. Usually, people broadcast messages only to announce that the machine is going down for maintenance, or for other important messages that affect everyone using the machine.

Type `wall` followed by Return. Then, type the text of the message, followed by Ctrl-D on a line by itself. The message appears on the screen — in the console window — almost immediately after you send it.

Figure 4-9 *Sending a Broadcast Message Using* `wall`

```
venus% wall
This machine will go down for maintenance at
noon today.     (Type Ctrl-D on next line to end text, send message.)
^D
Broadcast Message from venus!medici (ttyp4) at 12:00 ...

This machine will go down for maintenance at
noon today.

venus%
```

The same message appears on the screen, or console window, of anyone else who is logged in to that machine.

For more information on `wall`, see the `wall` Man Page, online or in the *SunOS Reference Manual.*

4.4. System Messages

System messages are like broadcast messages, only the system generates them automatically to notify you about something that may be important. One common system message is the *message of the day.*

When you log in, you often see two system messages — one about the operating system, the other about new mail — shown here as examples:

Figure 4-10 *Example System Messages*

```
venus login: medici
Password:
Last login: Fri Oct 31 23:59:59 from console
SunOS 4.2 Release 4.0 (DIONE_CLIENT) #1: Fri Feb 14 00:00:01 PST 1986

You have mail.
venus%
```

5

Other Features

5

Other Features

A variety of commands can help you read and send mail and messages.

5.1. Mail From Whom? The `from` Command

When you want to know whom your mail is from, without reading it using `mail` or Mail Tool, type `from` to your command prompt. For each letter waiting in your mailbox, `from` displays `From` followed by the sender's username, and the date and time it arrived in your mailbox.

Figure 5-1 *Who's My Mail From? The* `from` *Command*

```
venus% from
From sappho@aphrodite Sun Apr 1 8:45:12 1985
From rimbaud@verlaine Sun Apr 1 8:45:22 1985
From michaelangelo@david Sun Apr 1 8:45:45 1985
venus%
```

For more information on `from`, see the `from` Man Page, online or in the *SunOS Reference Manual.*

5.2. Who's Logged In On This Machine? `users`, `who`, and `w`

When you want to find out who's logged on to your machine, you can use one of three commands: `users`, `who`, and `w`.

The `users` command displays, in alphabetical order, the username of each person logged in on your machine.

Figure 5-2 *Who's Logged In On This Machine? The* `users` *Command*

```
venus% users
medici rimbaud
venus%
```

The `who` command provides more information than `users` does. For each terminal running on your machine, `who` displays the username, the terminal name, and the date and time you created the terminal *process.*

Figure 5-3 *Who's Logged In On This Machine? The* `who` *Command*

In the SunOS operating system, *processes* execute commands. The process that supports a terminal may run without any actual piece of hardware — what we usually think of as a terminal — associated with it; each window on a Sun Workstation counts as a separate terminal.

```
venus% who
medici   console Apr 1 8:50
medici   ttyp0   Apr 1 8:51
medici   ttyp1   Apr 1 8:51
rimbaud  ttyp2   Apr 1 9:36   (verlaine)
venus%
```

When a user has logged in to your machine from another machine, the name of that machine appears enclosed within parentheses after the rest of the information `who` displays about them.[36]

Rebooting a machine is, essentially, starting its software up. For example, you reboot when you turn the power on. You also reboot after your machine "crashes."

The `w` command gives yet more information. First, `w` displays system information, including the current time, how long since the last *reboot* of your machine, the number of terminals running on the machine, and system load information.

For each terminal running on your machine, `w` displays the username, the terminal name, the time of terminal login, other system information, and what program that process is running.

Figure 5-4 *Who's Logged In On This Machine? The* `w` *Command*

Note: The line in the figure starting with `-Ws` wrapped around, continuing from the end of the previous line.

```
venus% w
  9:43am  up 11:11, 4 users,  load average: 0.76, 0.45, 0.27
User     tty       login@  idle   JCPU   PCPU  what
medici   console   8:50am  4.02   4:40   3:59  clocktool -Wp 120 120
 -Ws 122 55
medici   ttyp0     8:51am     2   5:34   1:14  vi sculptor.list
medici   ttyp1     8:51am 94:14     15     15  date
rimbaud  ttyp2     9:36am      1      5      5  -csh
venus%
```

For more information on `users`, `who`, and `w`, see the appropriate Man Page, online or in the *SunOS Reference Manual.* To find out about some similar commands which give you information about users on *other* machines, see *Using the Network: Beginner's Guide.*

5.3. Who's Logged On Other Machines? Using `rsh`

One way to find out who is logged in on other machines within your local network, is to use the `users`, `who`, or `w` command in cooperation with the `rsh`, or remote shell, command.

Type `rsh`, followed by the *machine name* and *command* that you want to use. For example, when user `medici` on machine `venus` wants to find out lots of information about the users on machine `rose` (within `medici`'s local network):

[36] For more information, see the chapter about login access to other machines in *Using the Network: Beginner's Guide.*

Figure 5-5 *Finding Out Who's Logged In On Other Machines:* `rsh`

Note: The `rsh` command may take a little while on loaded machines or networks.

Note: The line in the figure starting with `-Ws` wrapped around, continuing from the end of the previous line.

```
venus% rsh rose w
  9:52am  up 2:36, 5 users,  load average: 0.67, 0.49, 0.33
User     tty       login@  idle   JCPU   PCPU  what
stein    console   8:50am  9:21   3:40   3:92  clocktool -Wp 120 120
 -Ws 122 55
stein    ttyp0     8:51am     2    :43    :14  talk toklas@rose
stein    ttyp1     8:51am 94:14     12     12  date
toklas   ttyp2     9:36am     1     10     10  talk stein@rose
woolf    ttyp3     9:40am  2:15     22     22  vi room
venus%
```

For more information on `rsh`, see *Using the Network: Beginner's Guide* and the `rsh` Man Page, online or in the *SunOS Reference Manual.*

5.4. The `finger` Command

Another command similar to `w` is `finger`, which gives you information about specific users. To find out more about this command, see *Using the Network: Beginner's Guide.*

5.5. The `vacation` Program

You still receive mail when you're gone — even if your machine is turned off. The `vacation` utility automatically sends a pre-written response to anyone who sends you mail. Incoming mail is not affected; `vacation` acts like an electronic mail equivalent of a telephone answering machine.

Simply type `vacation` to start the program. It will help you create the file which contains the automatic reply. This file is called `.vacation.msg` and lives in your home directory. `vacation` automatically sets you up in your normal editor to edit a standard version of the reply letter.

You can modify `vacation.msg` to say whatever you like. It should, however, start out with a `Subject:` line. If you include the word "$SUBJECT" in your reply, the subject of whatever letter you're replying to will be inserted at that point. Here's a sample `.vacation.msg` file:

Figure 5-6 *A Sample* `.vacation.msg` *File*

```
Subject:  I'm Away On Vacation.
Thanks for sending me your recent mail about $SUBJECT.
Currently I am in French Lick for the Indiana State
Free Throw Championships.  During this time, refer all
calls to my cat, Alfred.  I will be back on the 17th
of July.

Cosimo de Medici
```

This is how `vacation` works (in this case, using `vi`):

Figure 5-7 *Using* `vacation`

```
venus% vacation
This program can be used to answer your mail automatically
when you go away on vacation.
You need to create a message file in
/home/venus/medici/.vacation.msg first.
Please use your editor (/usr/local/vi) to edit this file.
     (Here you edit the sample vacation.msg)
You have a message file in /home/venus/medici/.vacation.msg.
Would you like to see it? n
Would you like to edit it? n
To enable the vacation feature a ".forward" file is created.
Would you like to enable the vacation feature? y
Vacation feature ENABLED. Please remember to turn it off when
you get back from vacation. Bon voyage.
venus%
```

To turn `vacation` off, or to modify your automatic reply letter, type `vacation` as you did to start it up.

`vacation` waits a specified interval before sending out your reply to someone it's already replied to; that way, someone who writes you several times while you're gone doesn't get your letter over and over again. (This specified interval is usually one week, but you can change it.)

Saving Mail with the `.forward` File

As shown in Figure 5-7, `vacation` creates a file called `.forward`. This file is one line long and looks like this:

```
\user,  "|/usr/ucb/vacation  user"
```

Mail programs look in `.forward` to see where they should send mail addressed to you. In the case of the `.forward` file shown above, mail is sent to the user *user* and to the vacation program. You can modify `.forward` (which lives in your home directory) with an editor; you could, for example, forward copies of all your letters to another user or another machine. One of the most common ways people use `forward` is to send copies of every letter they receive into a storage file. In Figure 5-8, the file `inbox` in the directory `/home/venus/medici` gets a copy of all incoming mail. (Note that it doesn't send mail to `vacation`, although it could.):

Figure 5-8 *Storing Incoming Mail*

```
venus% cat .forward
\medici, /home/venus/medici/inbox
venus%
```

Note! If you do forward mail to a file like `inbox`, be sure to prune it from time to time — it can get quite huge!

Complete information on `vacation` can be found either by reading the `vacation` section in the *SunOS Reference Manual* or by typing `man vacation`; information on the `.forward` file can be found under *aliases* in the same way.

5.6. Giving Your Real Name

That finishes the description of other useful commands associated with mail and messages. There is one more helpful thing for you to know, however. If you've been getting letters, you've probably noticed that some of them start something like this:

Figure 5-9 *A Letter Header*

```
From finches@galapagos Sun Feb 12 17:02:36 1859
From finches@galapagos (Charles Darwin)
To: medici@venus
Subject:  Huxley's 'The Descent of Bulldog'
```

You may be wondering how Mr. Darwin got his name in there. His username is *finches*, and his machine's name is *galapagos*. So where did "Charles Darwin" come from?

There is a file, called `passwd` (note the spelling), in the directory `/etc`. Among other lines, it has a line which has your username, your encrypted (ie., coded) password, and your real name, plus some other things. These pieces of information, seven in all, are set apart by colons and are called *fields*. The fifth field is the one with your real name in it. (It can contain all sorts of other information, such as your office or extension number, or job title.)

Figure 5-10 *An `/etc/passwd` File*

```
nobody:*:-2:-2::/:
daemon:*:1:1::/:
sys:*:2:2::/:/bin/csh
bin:*:3:3::/bin:
uucp:*:4:4::/var/spool/uucppublic:
news:*:6:6::/var/spool/news:/bin/csh
tecun:9So3iP:1897:10:Tecun Uman [Maya Quiche]:/tmp:/bin/csh
flann:slm0sms3:1896:10:Myles na Gapoleen:/tmp:/bin/csh
finches:0k3Xls:1901:10:Charles Darwin:/home/land/finches:bin/csh
```

Type `man 5 passwd` to see the layout of an `/etc/passwd` file. Or see the *passwd* entry in the *SunOS Reference Manual.*

If you want to change your name to include some pertinent information about yourself, you can, by modifying that name field. Charles Darwin's reads just "Charles Darwin," but if he changed it to "Charles Darwin, famous biologist," his letter would read like this:

Figure 5-11 *A Modified Letter Header*

```
From finches@galapagos Sun Feb 12 17:02:36 1859
From finches@galapagos (Charles Darwin, famous biologist)
To: medici@venus
Subject:  Huxley's 'The Descent of Bulldog'
```

Be careful to *not* use parentheses in your name field. Parentheses are automatically added by the mail system.

To modify the `/etc/passwd` file, you must become `root` on your machine. See the manual *Doing More with SunOS: Beginner's Guide* for information on how to do this.

6

Mail Over Networks

6

Mail Over Networks

This chapter describes remote networks for the purpose of understanding how to send mail across them. To find out about sending mail on your local network, see Section 1.2. For more information about networks in general, see *Using the Network: Beginner's Guide*.

6.1. What Is a Remote Network?

Earlier in this manual, you probably read the description of a remote network as a network that doesn't include, at least directly, the machine of the user to whom you're trying to send mail.

There are many different kinds of networks, each of which has a different syntax for the mail address of letters you want to send to the users on those networks. Some networks aren't connected to your network at all, so it is impossible to send letters to people on those networks.

6.2. What Networks Are Out There?

For the most part, you are likely to encounter people you'd want to send mail to on networks based on two major technologies:

- UUCP
- ARPANET/MILNET

Before attempting to send mail to someone on a remote network, you must find out which network they're on.

The UUCP Network

UUCP is a program which allows machines to use telephones to transmit data.[37] You send mail to other users by sending it through intermediate machines; each machine-to-machine pathway is unique. UUCP can be used to communicate with machines across the United States and throughout the world.

[37] UUCP is also a set of communications protocols.

Sending Mail to People on the UUCP Network

To send mail to someone on the UUCP network, you must know the *network path*, or sequence of machines the letter must travel through to get from your machine to the recipient's machine.

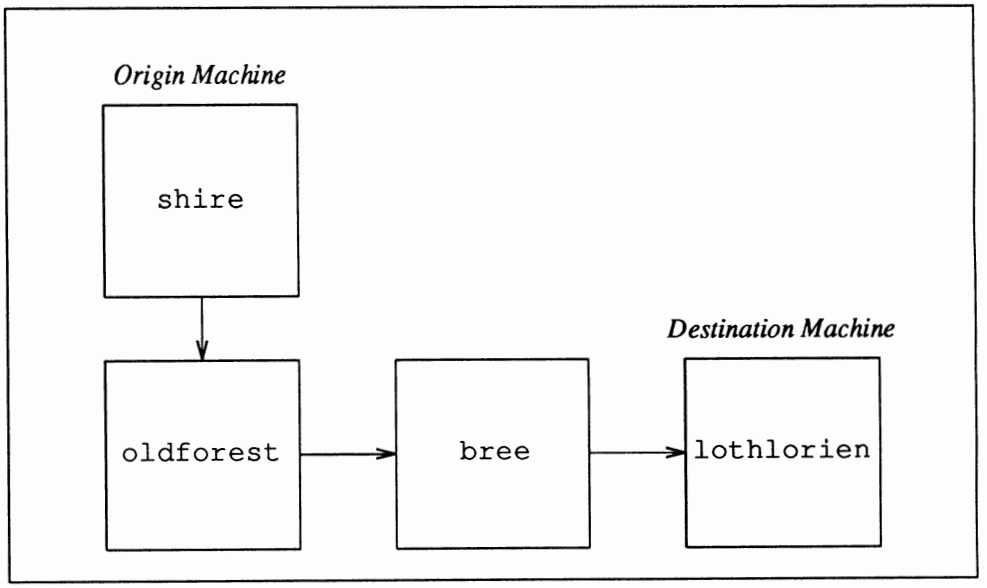
To find out machine name sequences necessary for mail addresses, ask prospective letter recipients if they know the appropriate network path. At the least, find out the prospective mail recipient's username and machine name.[38]

You can figure out the prospective recipient's mail address from this sequence of machine names. Pretend to walk along the path between the two machines starting with the first machine in the sequence and separating each *machine name* with an exclamation point (`!`), also called "bang.". Add the recipient's *username* to the end of the address after one last exclamation point.

Note: UUCP mail addresses may get quite lengthy.

For example, to figure out the mail address that user `bilbo` on machine `shire` would use to send mail to user `galadriel` on machine `lothlorien`, walk from `shire` to `lothlorien`.

Figure 6-1 *Sequence of Machines in Network*



The sequence of machine names is: `oldforest`, `bree`, and `lothlorien`. The recipient's username is `galadriel`.[39] So the complete mail address is:

```
oldforest!bree!lothlorien!galadriel
```

When you specify the mail address on the command line after `mail`, make sure to put a backslash character (\) before each occurrence of an exclamation point

[38] When the letter recipient doesn't know the appropriate mail address, ask your system administrator, if you have one. The system administrator may know offhand, or may have a map of the network.

[39] These names and places come from J.R.R. Tolkien's *Lord of the Rings*.

(`oldforest\!bree\!lothlorien\!galadriel` in the above example), so that the shell interprets the address properly.[40] However, it is not necessary to use a backslash when you're *already in* `mail` or Mail Tool. Backslashes are only needed when you're typing an address in as part of a command line.

You can learn about aliasing a mail address to another character string in the `mail` Man Page, online or in the *SunOS Reference Manual.*

How Does Someone Send Mail to Me on the UUCP Network?

When people with accounts on a UUCP machine ask you how they can send mail to you, try to come up with the appropriate network path. Determine your username, your machine name, and other machines you know your machine talks to using UUCP. Determine the other person's username, machine name, and associated machines. Hopefully, you will discover an associated machine in common, so that you can identify a network path between you.[41]

For more information on the UUCP network, see your system administrator, or look in *System and Network Administration.*

The Defense Data Network, or ARPANET/MILNET

The Defense Data Network includes two networks, the ARPANET and the MILNET, which are based on technologies developed by the Advanced Research Projects Agency of the U.S. Department of Defense. These are not classified, defense networks, however.

Sending Mail to People on the Defense Data Network

To send mail to someone on the Defense Data Network, you must find out the username and machine name of the mail recipient, usually by asking the recipient. Unlike the UUCP network, however, you don't need to know the names of all the machines between your's and the recipient's machine. The Defense Data Network takes care of that part automatically.

Construct the mail address by typing the recipient's *username*, followed by an at-sign character (`@`), the recipient's *machine name*, his or her institution's ARPANET name, and one of the following suffixes:

`.COM`
: This suffix is used by commercial organizations which are linked to the network (ie., private businesses). The form would be *username@machine.business.COM.*

`.EDU`
: This is for educational institutions, chiefly universities, which are tied to the ARPANET network. The form would be *username@machine.college.EDU.*

---

[40] The shell usually interprets exclamation points as part of the history mechanism. Putting a backslash before each exclamation point requires the shell to interpret the exclamation points as regular characters, rather than as special history mechanism characters. See *Getting Started with SunOS: Beginner's Guide* and *Doing More with SunOS: Beginner's Guide* for more information about the history mechanism.

[41] Some sites support `uuname`, a program which lists the names of systems accessible by UUCP, and `uupath`, which gives UUCP paths between known machines.

`.ORG`
This is used by non-profit agencies.

`.GOV`
Government agencies use this suffix.

`.MIL`
This is for military organizations.

Note that you do not have to capitalize the ARPANET suffixes — you can say, for example, `.com` instead of `.COM`. So, for user `lumpy` on machine `geewhiz` at Extreme South-Eastern Rhode Island University, the appropriate mail address would be:

```
lumpy@geewhiz.exsoeari.edu
```

How Does Someone Send Mail to Me on the Defense Data Network?

Providing your username and machine name should be sufficient for someone on a Defense Data Network machine to send you mail.[42]

For more information on the Defense Data Network, see *Using the Network: Beginner's Guide*, and the references to ARPANET/MILNET in the `sendmail` configuration guide, located in *System and Network Administration*.

[42] Sometimes, the situation gets more complex. Contact your system administrator, or look in *System and Network Administration*.

A

Modifying Mail and Mail Tool

A

Modifying Mail and Mail Tool

The `.mailrc` file contains a number of parameters which affect the way that `mail` and Mail Tool work. By changing this file you can customize these programs to behave the way you want them to. `.mailrc` is usually located in your home directory.

This is what a typical `.mailrc` file might look like:

Figure A-1 *A Typical* `.mailrc` *File*

```
set alwaysignore
set alwaysusepopup
set askcc
set autoprint
set bell=3
set flash=3
set folder=/home/venus/medici/mail
set hold
set metoo
set noeditmessagewindow
set onehop
set outfolder
set printmail='lpr -h -p'
set record=outbox
```

A.1. The Defaults Editor

A *default* is the automatic, assumed value of a setting; ie., the value it has if you do nothing. If, for instance, you normally leave your office door open, you can say that the default for the door is "open."

If you are running the SunView windows program, you should use the Defaults Editor to modify `.mailrc`.

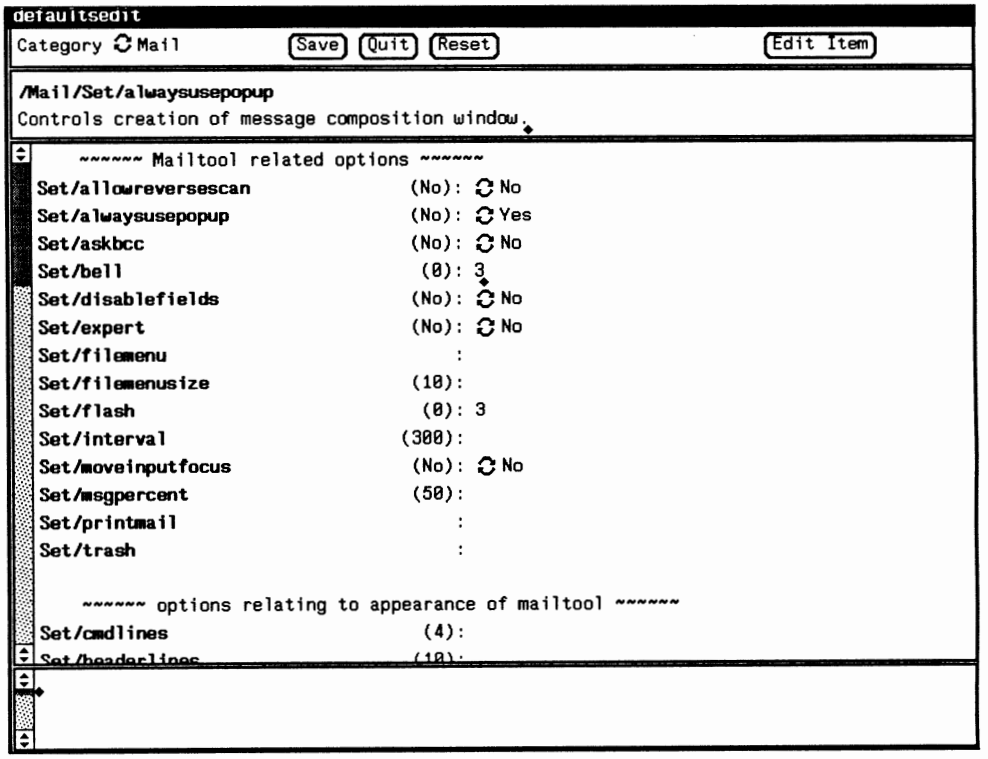
The Defaults Editor is a simple, interactive program which enables you to change a number of SunView defaults in addition to those in `.mailrc`. The Defaults Editor presents you with a number of settings you can change, with short explanations of each item. To bring it up, choose 'Defaults Editor' from the 'Editors' menu in SunView, or you can type as follows:

Figure A-2 *Bringing Up the Defaults Editor*

```
venus% defaultsedit &
venus%
```

The *SunView 1 Beginner's Guide* contains an explanation of how to use the Defaults Editor.

Figure A-3 *Changing* `.mailrc` *with The Defaults Editor*



With the Defaults Editor, many `mail` and Mail Tool options are turned on by setting them to 'Yes' and switched off by setting them to 'No'.

However, if you can't use the Defaults Editor, you can still modify `.mailrc` with a conventional editor such as `vi`. To set a `.mailrc` option, include a line like this:

`set` *option*

To turn an option off, either remove the above line, or put the prefix *no* in front of the option's name:

`set no`*option*

in `.mailrc`.

Some options in `.mailrc` are not simple "on/off" settings, but require you to input a value. For example, to set your folder directory, you type its name in at the appropriate place in the Defaults Editor. If you're not using the Defaults Editor, you'd put in a line like

```
set folder=/home/medici/mail
```

(or whatever your folder directory will be).

You can set a variable to 'No' *only* when you are using the Defaults Editor. Do not put a line in your `.mailrc` file which says `set option=No`.

Important Note: "Turning an option on," "setting an option," and using the Defaults Editor to set it to 'Yes' are all synonymous; "turning it off," "unsetting" it, and setting it to 'No' also all mean the same thing.

If you do not change a setting in `.mailrc`, either with the Defaults Editor or with another editor, then the default value is assumed.

There is a sample `.mailrc` file, called `Mailrc`, in `/usr/lib`. It contains many convenient option settings; to get a copy which you can use, copy it to your home directory as follows. (Before you do, type `ls ~/.mailrc` to see if you already have your own `.mailrc` file. If you get nothing back but your prompt, then you don't.)

Figure A-4 *Copying the Sample* `Mailrc` *File*

```
venus% cp /usr/lib/Mailrc ~/.mailrc
venus%
```

For more information on `mail` and Mail Tool, and their associated options, see the *SunOS Reference Manual.* (Mail Tool is spelled *mailtool.*)

Here are the various settings in `.mailrc`:

Mail Tool-Related Options:

allowreversescan:

When turned on, allows you to go through the letters in your mailbox in reverse order; ie., last to first. This affects which letter is *next* — if the sense of direction is reverse, then the letter displayed by the **Next** button is actually the *previous* one. Default setting: turned off.

alwaysusepopup

Turning this on makes the composition window come up as a separate window frame; otherwise the composition window is simply split off from the message subwindow. Default: turned off.

askbcc

This gives a "Bcc: " prompt when set. "Bcc" stands for *blind carbon copy*; it's the same as "Cc: " except that the list of people you copy the letter to doesn't appear in the letter's header, so you can copy a letter to someone without alerting the addressee. This sneaky little option in normally turned off.

bell

Number of times you want the terminal to beep when you get a letter. The default is no beep at all. See also `flash`.

disablefields

Including this option removes the fields in the composition window's address template. Default: turned off.

editmessagewindow

With `editmessagewindow` turned on, if you try to edit a letter in the message window, Mail Tool will first ask you to confirm that you want to edit it. Normally this is turned off.

expert

When you have `expert` set, Mail Tool does not ask you to confirm deletions, cancellations, etc. Normally turned off.

When you set a variable to have a value which is expressed by more than one word, put the variable in quotes. If you want `filemenu` to include more than one file, put the list of files in quotes, for example, `'+trash +mbox'`.

filemenu

The `File:` prompt in the command panel window has a menu associated with it, of folders you've been working with. You can set `filemenu` so that certain files are automatically included on this menu when Mail Tool starts. Examples include `+trash` and `+mbox`. (See `trash` and `MBOX`, below.)

filemenusize

Maximum number of files in the `File:` prompt's menu. The default is ten.

flash

Number of times to flash Mail Tool when mail arrives. Also flashes the Mail Tool icon when Mail Tool is closed. The default is zero.

Mail Tool cannot flash without beeping, but it can beep without flashing. This means that if you set `flash` to 3 and `bell` to 1, you will get one flash and one beep. If you set `bell` to 3 and `flash` to 1, you will get three beeps and one flash. See `bell`, above.

interval

Time that Mail Tool waits before checking for new mail (in seconds). The default is 300 seconds (5 minutes).

moveinputfocus

`moveinputfocus` controls where you type when you start a composition window: with `moveinputfocus` set, the composition window automatically becomes the window you're typing in as soon as it comes up. This feature only has meaning if you are using "Click-to-Type," described in the *SunView 1 Beginner's Guide*. Normally turned off.

msgpercent

This controls how much of the message subwindow will be used for a composition window when composing a letter. Normally set to fifty percent. If `alwaysusepopup` is set, though, this setting has no meaning.

Remember that multi-word options go in quotes; if you change `printmail` to (say) `lpr -h`, you have to write '`lpr -h`'.

printmail

The command for printing a letter. You can use whatever printing scheme works best for you. Normally set to `'lpr -p'`.

trash

`trash` is a file which collects your deleted letters; they stay here until you push the **Done** button. Setting `trash` allows you to look at deleted letters as though they were saved in a regular file. You set `trash` to the name of your trash file. If set to `+trash`, you can access like any other folder.

Options Relating to the Appearance of Mail Tool:

headerlines

Size of the header list subwindow. Default: ten lines.

maillines

Size of the message subwindow. Default: thirty lines.

popuplines

This sets the size of pop-up composition windows. Default: set to 30 lines.

Options Affecting Both Mail and Mail Tool:

allnet

All network names whose username (ie., `helen` in `helen@troy`) match are treated as identical. Default: turned off.

alwaysignore

See Section A.3 for more on `alwaysignore`.

append

Normally letters are added to the end of `mbox`; if you prefer to have your most recent arrivals go to the beginning, set `noappend` or set this to 'No' with the Defaults Editor. See `MBOX` and `hold`.

ask

No longer implemented. See `asksub`.

askcc

When this is set you are automatically given the "Cc: " (carbon copy) prompt when composing a letter. The default setting is to have this feature turned off.

asksub

With this set you're automatically prompted for a subject when composing a letter. By default this feature is turned on.

autoprint

With `autoprint` on, `mail` and Mail Tool display the next letter in the mailbox when one is deleted. `autoprint` is turned off by default.

DEAD

This ghoulish variable takes the name of a file (with its full path name) where partial letters get stored in case of an interruption like a power failure. `DEAD` is normally set to be a file called `dead.letter` in your home directory. The `save` variable must be set for this variable to take effect. See `save`.

folder

This is the directory which contains your mail folders. For more information, see the `outfolder` variable below and Section 2.10.

hold

When `hold` is turned on, letters which you've read are still kept in your mailbox until you save or delete them. When `hold` is turned off, already-read letters are moved to a file, usually located in your home directory and called 'mbox', when you do a commit. `hold` is turned off for `mail`, on for Mail Tool. See `MBOX`.

indentprefix

When composing a letter, you can include another letter, indented to set it off. `indentprefix` is what gets put to the left of a letter when it's indented. The default is just a tab; you can put in one or more characters of your choice, surrounded by quotes, to indicate that this is an included letter.[43]

keep

`keep` signals that you want to keep your mailbox even when it's empty. Turning `keep` on means that your mailbox is truncated to zero length when empty, instead of removed and created anew when you get mail. Default: turned off.

keepsave

Normally when you save a letter into a file or folder, you delete it from your mailbox.[44] Setting `keepsave` prevents `mail` or Mail Tool from deleting it automatically.

Remember, multi-word variable values, like `ls -F`, should go in quotes.

LISTER

`LISTER` is set to a SunOS shell command which you use for displaying the contents of your folder directory. In Mail Tool, the default is '`ls -F;`' in `mail`, the default is `ls`.

[43] A tab is indicated by a Ctrl-I or `^I`.

[44] In Mail Tool, letters are not deleted if you use the 'Copy' option of **Save**.

If you change `LISTER`, the command you replace it with must display directories as a "/" the way '`ls -F`' does, for Mail Tool to work correctly. See `ls` in the *SunOS Reference Manual*or type **`man ls`** for more information.

MBOX

Normally, letters in your mailbox which you've read are kept there until deleted or saved. When `hold` is turned off, however, these letters are saved into a file specified by `MBOX`. Normally this is a file called `mbox` in your home directory. See `hold`.

metoo

When you send something to an alias group of which you're a member, you don't receive a copy of the letter unless you specifically address it to yourself as well. Setting `metoo`, however, includes you among the recipients. Default: turned off.[45]

onehop

When you receive a letter that was sent to several people, the other recipients' machine addresses are normally given relative to the *author's* address. Setting onehop allows the others' addresses to be given relative to your own — ie., just "one hop" away from you, not through the author. This makes your replies more efficient.

outfolder

You can keep a record of every letter you send; they go into a file set by the variable `record`. If `outfolder` is turned on, then this file will be located in your folder directory. By default, `outfolder` is turned off. See also `folder` and `record`.

record

You set `record` to be the name of a file which receives a copy of every letter you send. If `outfolder` is turned on, then this file is located in your folder directory (set with the `folder` variable.) If `outfolder` isn't turned on, then `record` should include the full pathname of this file.

replyall

The net effect of setting `replyall` is to reverse `mail's` **R** and **r** commands, or, in Mail Tool, to reverse the meanings of 'Reply' and 'Reply (all)'. The default setting is to have `replyall` turned off.

save

When turned on, `mail` and Mail Tool save partial letters into the file specified by `DEAD` in case of an interruption like a power failure. Turned on by default; see `DEAD`.

---

[45] `metoo` will work for alias groups which you declare in your `.mailrc` file. Whether it works for other mail aliases depends on how your system is set up.

sendmail

`mail` and Mail Tool usually use the program `sendmail` to deliver mail; you can specify an alternate program here. Appendix B gives an example of a shell script which formats a letter before sending it.

showto

Sometimes you send letters to yourself (for example when you Cc: yourself or send a letter to an alias group which includes you). If you set `showto`, `mail` and Mail Tool display the letter's recipient, rather than the sender (who is yourself). That way you see why you received mail you sent, rather than that you sent it. By default, `mail` and Mail Tool display the sender in all cases.

Options That Affect Only Mail:

bang

Enables the special-casing of exclamation points (!) in shell escape command lines as in `vi`. Default is turned off.

cmd

The default shell command for the `pipe` command in `mail`. (For advanced users.)

conv

Convert uucp addresses to the specified address style. The only valid conversion now is `internet`, which requires a mail delivery program conforming to the RFC822 standard for electronic mail addressing. Conversion is normally disabled. See the **-U** command-line option in the `mail` section of the *SunOS Reference Manual.*

crt

`crt` is a number roughly corresponding to the number of vertical lines in many terminal screens. If a letter has more than this number of lines, `mail` pipes it through a displaying command set by `PAGER` (usually the `more` command). Default: turned off. See `PAGER`.

dot

Accept a dot ('.') alone on a line to terminate a letter. This is the default.

EDITOR

The `edit` and `~e` commands invoke an editor to use when writing letters; `EDITOR` sets the editor to use. The default is `ex`.

escape

You can replace the `~` in commands such as `~h`, `~e`, and `~m` with a character of your own choosing.

header

This variable is normally set, so that when you enter `mail` the header list is displayed. You can suppress the initial display of the header list by turning this variable off.

ignore

You may wish to ignore a Ctrl-C when typing a letter, especially if you have a noisy dialup line. Normally this is set so that you notice it.

ignoreeof

Normally you terminate a letter with a Ctrl-D. Setting `ignoreeof` means you must end a letter with either a period on a line by itself or the `~` command.

page

Insert a form feed after each letter. Default: turned off.

PAGER

The command which `mail` uses to display long letters; usually `more`. See `crt`.

prompt

The `mail` prompt. Usually `&`; you can set it here.

quiet

`mail` normally displays a short letter, including its version number, when it starts up. You can turn this off to suppress this letter.

screen

The maximum number of header summaries to be displayed at one time; ie., the amount of the screen to be taken up by the header list. No default.

sendwait

With `sendwait` set, `mail` (or Mail Tool) waits until it has finished sending off a letter before coming back to the user. Turned off by default.

SHELL

The name of a preferred command interpreter; usually *sh*; you can set this to (for example) */bin/csh*. This is inherited from the environment unless you specify it here..

Remember to enclose your signature in quotes.

sign

You can "sign" your letter with the `~a` command in `mail`; the `sign` variable is your "signature." It could be some pithy phrase you want to finish your letters with.

toplines

The `top` command in `mail` prints out the first few lines of letters whose letter numbers you give. `toplines` specifies how many lines to print out. Default: five lines.

verbose

When set, `sendmail` is used with the `-v` (`verbose`) option. Normally not set. See `sendmail (8)` in the *SunOS Reference Manual* or type **`man 8 sendmail`**.

VISUAL

The name of the screen editor used when you type the `~v` command in `mail`. Default: `vi`.

A.2. Aliases

Additionally, you can put *aliases* in your `.mailrc` file. An alias is a group of user names all grouped under a single name. For example, if you wanted to send mail on a regular basis to `joe@donkey`, `bill@whitehouse`, and `laura@smiley`, you could enter this line in `.mailrc`:

Figure A-5 *A* `.mailrc` *Alias Group*

```
alias buddies joe@donkey bill@whitehouse laura@smiley
```

and you send them letters just by sending a letter to `buddies`.

Two things to note about aliases in your `.mailrc`: First, don't separate the names of their members with commas. Just use spaces. Second, don't specify a machine name when you mail to the alias; just give the name of the alias.

You cannot currently set up aliases with the Defaults Editor; you must edit `.mailrc` with an editor such as `vi`.

Local Alias Groups — `/etc/aliases`

While we are on the topic of aliases, we should mention the `/etc/aliases` file, which allows you to set up a *local alias group*. This is similar to the `buddies` alias we set up in section A.2, but it differs in this respect: only you can use an alias group (like `buddies`) in your `.mailrc` file, whereas other people can use an alias group which you declare in `/etc/aliases`.

Suppose you're user `medici` on the machine `venus` and you put this into your `/etc/aliases` file:

Figure A-6 *A* `/etc/aliases` *Alias Group*

```
lunchers: joe@donkey, bill@whitehouse, laura@smiley, tecun@uman,
          flann@swim2birds, odysseus@ithaka, weasel@toadhall
```

Now anyone who sends mail to `lunchers@venus` will send mail to everyone on `lunchers` mailing list. (Unlike the aliases in `.mailrc`, you must give the

name of the machine that has the altered `/etc/alias` file. Also, note that the members of the alias *are* separated with commas.)

You must become `root` to modify your `/etc/aliases` file. (Becoming `root` is explained in *Doing More with SunOS: Beginner's Guide*.) There's a lot more you can do with `/etc/aliases`; for more information, see *addresses* in the *SunOS Reference Manual*, or type **`man aliases`** or **`man addresses`**.

A.3. Suppressing Header Lines in Letters

You've probably noticed that your mail letters start out with a header which contains lines similar to these:

```
From loeb@leopold Fri Aug 21 15:18:42 1924
Return-Path: <loeb@leopold>
Received: from leopold.XXX.uucp by darrow.XXX.uucp (3.2/SMI-3.0)
    id AA21411; Fri, 21 Aug 87 15:18:32 PDT
Received: by leopold.XXX.uucp (3.2/SMI-3.0)
    id AA20410; Fri, 21 Aug 24 15:22:11 PDT
Date: Fri, 21 Aug 24 15:22:11 PDT
From: loeb@leopold (Arthur Garfield Hayes)
Message-Id: <8788212322.AA00410@leopold.XXX.uucp>
To: clarence@darrow
Subject: Scopes Case Background
Status: RO
```

You can have `mail` and Mail Tool suppress the display of any of these lines by including an "ignore" line in your `.mailrc` file. For example, the following line

```
ignore message-id return-path received date status via
```

produces a header which looks like this:

```
From loeb@leopold Fri Aug 21 15:18:42 1924
From: loeb@leopold (Arthur Garfield Hayes)
To: clarence@darrow
Subject: Scopes Case Background
```

You can pick and choose which of the header categories you want to have displayed.

You cannot currently use the Defaults Editor to add an ignore line to `.mailrc`. You must edit `.mailrc` with a conventional editor such as `vi`.

The header categories which you tell `mail` and Mail Tool to not display are still included when the letter is saved. However, if `alwaysignore` is set, then these header lines are not saved, either. This is also true for copying and including letters as well.

In Mail Tool, you can use the **Show** button's menu to override the `ignore` option. The 'Show Full Header' option will display a letter's full header.

B

Formatting Letters

B

Formatting Letters

There is a SunView feature called "word wrap" or "Wrap at Word"; it allows you to type in a text window without having to end your lines with a carriage return. It's quite useful, for example, when you're modifying an existing line: it automatically reformats your text as you do insertions and deletions. Line wrap is discussed in the *SunView 1 Beginner's Guide* in the section on the Text Edit menu.

Using word wrap in Mail Tool can be problematic, however, because not all the people running Mail Tool use word wrap. This means that a perfectly fine-looking letter may come out with overlong lines when viewed by its recipient. Such a letter may have lines which disappear out of the letter window or may appear with lines chopped in the middle of words. You can pretty up such a letter that *you* receive by using the 'Fmt' option of the SunView Text Edit menu. 'Fmt' (which invokes the `fmt` command) formats your lines to a standard width. For more about 'Fmt' see the *SunView 1 Beginner's Guide* and `fmt` in the *SunOS Reference Manual.* (Or type **`man fmt`**.)[46]

There is a way to use `fmt` to format letters you send every time you use `mail` or Mail Tool, regardless of whether you are running the SunView windows system. That way you can enjoy the benefits of word wrap without having to worry how your mail will look when viewed by its recipient. The `sendmail` variable in your `.mailrc` file controls what program you use to send mail. Normally it's set to the program `sendmail`, but you can set it to be a shell script which formats your letter before sending it on to the `sendmail` program.[47]

`fmt_mail`, located in `/usr/bin`, is such a shell script. First, set *sendmail* to "fmt_mail." Then copy `fmt_mail` into a directory that's contained in your `path` variable, like your home directory or your private `bin` directory.[48] When you send off a letter, `fmt_mail` will format your letter, and then mail it using the `sendmail` program.

---

[46] (Besides using `fmt` from the Text Edit menu, you can map it to one of your function keys. See the section on mapping to function keys in the *SunView 1 Beginner's Guide.*

[47] See Appendix A for more on `.mailrc`, the Defaults Editor, and *sendmail.*

[48] To see what your path is, check your `.cshrc` file in your home directory.

C

Glossary

C

Glossary

This glossary lists mail and message terms in common use, especially in this manual.

abort a letter
To decide not to send a letter and to interrupt the mail sending process.

active letter
In Mail Tool, this is the letter in the header list which you delete, save, reply to, and so on. The active letter is selected by moving and clicking the mouse. This letter may or may not be the *current* letter, ie., the one displayed in the message window.

broadcast message
Message sent to all users on a machine

button
Either:

- One of the three buttons that you can press on a mouse.
- "Software" representations of buttons on a control panel within a window and mouse program like Mail Tool.

click
To press and release a mouse button.

command panel
With Mail Tool, a section of the window that presents the command buttons.

compose
Create a letter.

console
A terminal, or a special window on the screen, where system messages appear.

current letter
In Mail Tool, the letter displayed in the message window; this may be different from the *active letter*. In `mail`, the letter that you last read, or the first letter that `mail` pulls from your system mailbox when obtaining new mail.

electronic mail
The same as **mail**.

electronic messages
The same as **messages**.

field
A place in a Mail Tool composition window for entering data, such as the recipient's name, the subject of the letter, and so on. The *active* field is darkened and is replaced by your input. Fields are delimited by |> and <|, and you move between them by typing Ctrl-Tab.

folder
A file, containing letters, located in your *folder directory*. A folder is specified by preceding its name with a plus sign (+). Folders are useful because they save typing.

folder directory
A place to group together your files containing mail. The folder directory's name can be represented by a plus sign (+), so that a file called `new_mail` in your folder directory can be written as `+new_mail`.

gateway
A link between two networks.

header
The part of a letter that contains information about the letter, such as the subject and recipient.

header list
A list of header summaries. Each header summary gives information about one letter; the header list displays summaries of all the files in your mailbox, or in a file or folder. In Mail Tool, the header list is displayed in the header list window; in `mail`, it's displayed with the **h** command.

header summary
See *header list*.

icon
A small rectangular window on the screen that identifies a closed, or iconic, window and mouse program.

interactive message
A message that someone can read and respond to while you, the message sender, wait.

local network
A network of machines directly connected with your machine that may communicate (through a gateway) to remote networks.

mail
Electronic correspondence from one user to another on a computer network.

mailbox
A file which receives and stores new mail. Also known as the *system*

mailbox.

menu
With some window and mouse programs, a selection of possible action choices presented in a rectangular box. You pick items with the mouse.

menu item
One of the possible choices on a menu.

letter header
The same as **header**.

letter number
The same as **number**.

letter status
The same as **status**.

letter text
The same as **text**.

messages
Immediate and interactive electronic communication between users on a local computer network, not to be confused with a **mail** message.

network
Technically, the hardware connecting various machines, allowing them to communicate; informally, the machines so connected. There are various kinds of networks.

network path
A series of machine names used to direct mail from one user to another.

new letter
A letter that you have just pulled out of your system mailbox.

number
A letter characteristic that allows you to choose that letter.

old letter
A letter that you've already read.

pipe
Software connection between two programs.

Postmaster
The username designated to receive notice of letters that the mail facility cannot deliver.

process
The operating system software SunOS uses to execute commands.

remote network
A network that doesn't include, at least directly, your machine.

ring a party
Try to attract the attention of the recipient of a `talk` message by displaying

messages on the recipient's screen.

select
Choose, usually by pressing mouse buttons.

shell script
A series of commands run as a program by the shell.

status
New, old, or unread classification of a letter.

subwindow
One of multiple windows within a window-based tool like Mail Tool.

system message
Messages that the system generates automatically to notify you about something that may be important, such as new mail, message of the day, or login information.

terminal
A process running on a machine that originates with the physical device called a terminal, or as a software representation of such a physical device, like a window.

text
The part of the letter that contains the communication to the recipient, to be distinguished from the letter header.

trash bin
A file in which `mail` and Mail Tool stores deleted letters, if so chosen.

unread letter
A letter that you already pulled from your system mailbox, but have not yet read.

window and mouse program
A program that makes use of SunView, the window system and associated software, so that you can use the mouse to locate and select items within the tool, all within a window on your screen.

window-based tool
The same as **window and mouse program**.

Yellow Pages
A directory of usernames and machine names on a local network that provides automatic machine name addressing of letters.

Index

Notes