



# Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA

RECEIVED  
LAWRENCE  
BERKELEY LABORATORY

## Computing Division

JAN 28 1985

LIBRARY AND  
DOCUMENTS SECTION

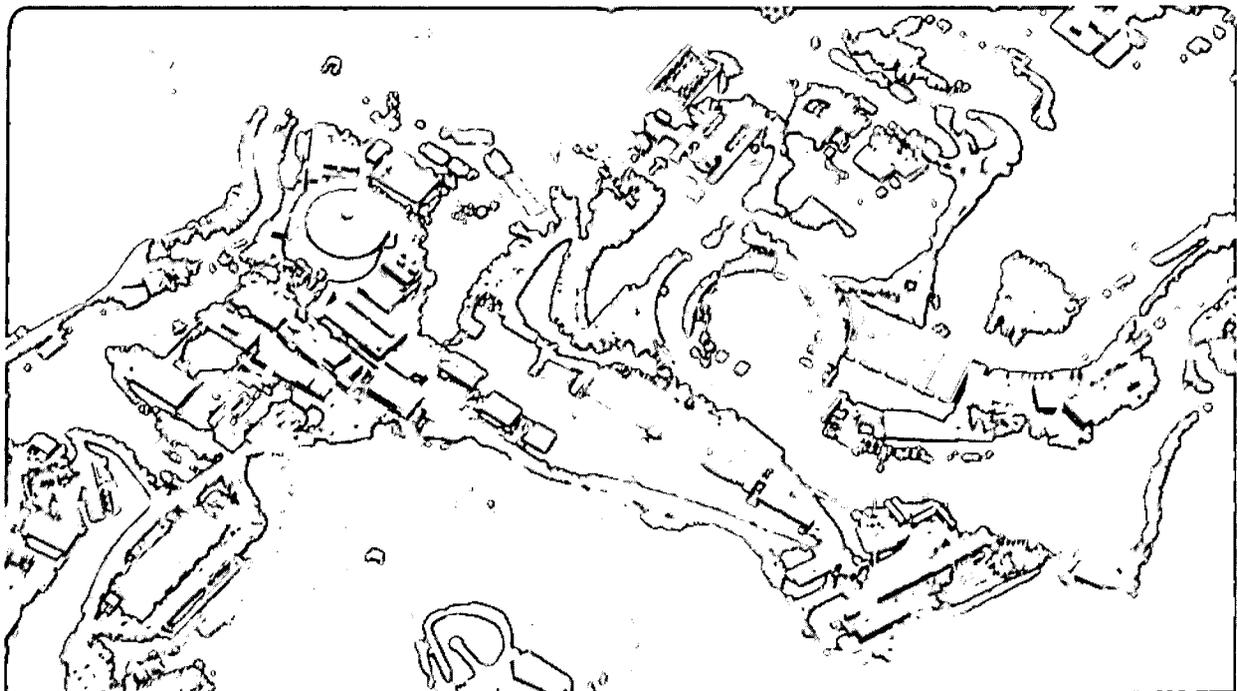
A TUTORIAL FOR THE SOFTWARE TOOLS MAIL SYSTEM--MSG

V.A. Sventek

October 1984

**For Reference**

Not to be taken from this room



PUB-3049  
c.1

## **DISCLAIMER**

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

---

# **A Tutorial for the Software Tools Mail System — MSG**

October, 1984

Virginia A. Sventek  
**Computer Science Research Department**

Computing Division  
Lawrence Berkeley Laboratory  
University of California  
Berkeley, California 94720

---

This work was supported by the Applied Mathematical Sciences Program of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098 and the U.S. Department of Labor, Employment and Training Administration.

# Table of Contents

---

<b>1</b>		<b>Overview</b>
	4	Using This Manual
<b>5</b>		<b>Starting MSG</b>
	5	Starting the Electronic Mail System
	7	How to Enter MSG Commands
	8	Getting Help
<b>11</b>		<b>Leaving MSG</b>
	12	The Exit Command
	13	The Quit Command
	14	The Overwrite Command
<b>15</b>		<b>Reading Messages</b>
	15	Message Sequences
	16	Message Headers
	18	Specifying a Message by Number
	21	Specifying a Message by Sender Name
	23	Specifying a Message by Subject
	24	Reading All Messages
<b>25</b>		<b>How to Tell What's in Your Mail Box — Headers</b>
	25	All Messages
	26	By Number
	27	By Sender
	27	By Subject
<b>29</b>		<b>Deleting Messages</b>
	30	By Number
	30	By Sender
	31	By Subject
	31	All Messages
<b>33</b>		<b>Recovering Deleted Messages</b>
<b>35</b>		<b>A Filing System for Messages</b>
	36	Creating File Folders
	38	Moving Messages Into Folders
	41	Accessing Messages Stored in Folders
	45	Viewing and Adding Folders

## Table of Contents, Continued

---

<b>47</b>		<b>Sending Messages</b>
	<b>48</b>	Filling out the Header — the Envelope
	<b>51</b>	Leaving Sndmsg — When You Do not Want to Send the Message
	<b>52</b>	Changing the Header
	<b>55</b>	Looking at the Header
	<b>57</b>	Composing the Message
	<b>60</b>	Sending the Message
	<b>61</b>	Incorporating VMS Files into Your Message
<b>65</b>		<b>Answering Your Mail</b>
<b>67</b>		<b>Forwarding Messages</b>
<b>71</b>		<b>Moving Your Mail to a File —     Getting a Paper Copy</b>
<b>75</b>		<b>Return Receipts</b>
<b>79</b>		<b>Creating and Using Mailing Lists, Malias</b>
<b>83</b>		<b>MSG Command Summary</b>
<b>87</b>		<b>Index</b>

---

## Overview

---

### Why Use Electronic Mail?

Over the past several years Electronic Mail has moved out of the arena of the computer specialist and into the mainstream of computer users. People in corporate offices and governmental agencies are discovering that the kinds of immediate and continuous communication that electronic mail systems provide greatly increases their productivity and improves their ability to share ideas and reach decisions. Electronic mail is a convenient way to put thoughts in writing and have them communicated to others. It encourages an informal style of writing, yet thoughts and decisions are communicated precisely and immediately, making it an ideal management tool.

There are several other compelling reasons for learning how to use an electronic mail system, the first being the speed with which messages can be transmitted. Because electronic mail is automatically sent across telephone lines that stretch across the country it can be read almost immediately.

Secondly, people aren't always in their offices when they are needed; by using electronic mail you can solve the expensive, and often irritating, "telephone tag" problem.

The third reason is that electronic mail makes it possible for people to maximize their computer learning experience by providing them with a tool that is easy to learn and at the same time is productive. Electronic mail can be of great assistance when you are preparing documents, computer programs and software procedures, and puts you in close contact with people who can answer questions and give you help while you are learning other tools.

The most compelling reason, though, is that electronic mail is easy to use. As long as you can type with one finger, you can handle your own correspondence tasks and be assured that your messages are delivered in the fastest possible manner.

## An Overview of Electronic Mail

When you are learning to do something new, it often helps to draw analogies or comparisons to those things with which you are already familiar. This is certainly true of electronic mail.

If you think about it, the way you send and receive mail is quite structured. This structure makes it possible to draw very clear comparisons between the way you handle mail now and the way you will work with it electronically.

### **Delivery**

With paper mail, someone delivers the mail to a central location. It's sorted and handed out to each person in the office. You look at your own stack and decide which piece you want to open first.

Electronic mail is automatically distributed across the network and makes its way through the network to your electronic "incoming mail box."

### **Filing**

After you've finished reading a piece of paper mail you can either file it somewhere or throw it away. (You probably have a filing system that makes it easy for you to lay your hands on a specific piece of paper.) So, whenever you get a new piece of mail you probably file it in the appropriate place, if you don't throw it away.

Electronic mail allows you to both file or throw away messages, and assists you in creating and maintaining individual file folders in which you can store messages.

### **Answering**

When you answer a paper letter, you usually add a sentence in the response that refers back to the original letter.

The electronic mail system includes such a reference automatically.

### **Forwarding**

If you receive a letter that you want someone else to read, you forward it, and include a line or two explaining why you've sent it. By the same token, if you want to share the message with others you make photocopies and pass them on.

You can forward any of your electronic messages to as many people as you need, and the mail system will automatically include a note explaining who forwarded the message.

## An Overview of Electronic Mail, Continued

### Return Receipts

Occasionally you will send important letters by registered mail. The Post Office notifies you of the letter's delivery.

Electronic registered mail goes one step further: it tells you the date and time the message was *read* rather than when it was delivered.

### Sending Letters

When you send a business letter to someone, you write it out or have it typed; then you address an envelope and put it in the mail box. If you are including attachments you put them in the envelope along with the letter.

When you send an electronic message you first fill out the envelope and then write the letter (using the editor that you feel most comfortable with). If you want to include attachments they can be added to the letter by including the contents of any file stored in your computer.

In short, electronic mail makes it possible for people who share the same office as well as those who are thousands of miles apart to work together, whether their aim is to manage projects or staff, write documents and reports, or develop computer programs or procedures. In essence, electronic mail makes it possible to think of all network users as working together in one large office.

## Using This Manual

This manual was written with two purposes in mind: (1) to get you started using the system quickly; and (2) to explain the more advanced features of the system you will need when your mail load gets heavier and your time to spend dealing with it gets shorter.

The first section shows you how to use the simplest form of each command, giving you just enough information to use the mail system comfortably and still be productive.

The second section defines additional terms and describes the more technical aspects of the system, and should be used as a reference guide.

You will find that the more you use the mail system, the more you will want to learn short cuts, and the reference section will provide you with the necessary information.

### Notations

Each example in the manual follows the same pattern: the things **you** are to type are printed in **bold face**; the responses made by the mail system are shown in pale face.

Whenever you see the symbol **RET** it means that you should hit the key labeled "Return" on your terminal.

The symbol **<Blank Line>** means that you should insert a blank line by hitting the **RET** key. The symbol **(Space)** means that you should hit the space bar once.

---

## Starting MSG

---

The electronic mail system described in this manual is called "MSG."

In order to use the MSG system, you must be logged onto the computer, and be at DCL level which means that you should see the \$ (dollar sign) prompt when you hit the **(RET)** key.

The example below shows you what you must type in order to log on the computer.<sup>1</sup>

```
(RET)

USERNAME: SMITH (RET)
PASSWORD: password (RET)

Welcome to LBLH-CSAM VMS 3.5 VAX 11/780

LBLH will be down on Thursday, September 6 for
preventive maintenance.

Type HELP if you need assistance.

$
```

---

<sup>1</sup> If you need help at this point, you should look at the booklet entitled "Introduction to DCN;" it contains a far more complete explanation of the log on procedure, including sections on how to turn on the terminal and use the special characters on your keyboard.

## Starting Msg, Continued

To start the MSG system, type the word "msg" when you see the \$ prompt.

As the MSG system starts up, it prints a "banner message" which introduces the system and tells you which characters you can type to get help.

Immediately after printing the banner message, MSG will print a list of the new messages that have arrived in your mail box since the last time you logged onto the machine. "Old" messages, that is the messages that you read the last time you were in MSG, will not appear in this list. To see their headers, you should refer to the section entitled "Headers — How to Tell What is in Your Mail Box."

When MSG is ready for you to type in commands, it will print the symbol <- at the left side of your terminal screen. This symbol is called the MSG Command Prompt and whenever you see it, you can enter MSG commands.

```
$ msg (RET)

Software Tools MSG System
type ? for help
type # for news
type % for intro

10 messages / 10 new
N      1  427  6-Jun-1984 virginia          vacation schedules
N      2  304  6-Jun-1984 virginia          maintenance
N      3  525  6-Jun-1984 schroeder        Using vtroff
N      4  347  6-Jun-1984 census           test
N      5  481  6-Jun-1984 nadel            Meeting to discuss contracts
NR     6  546  6-Jun-1984 mccarthy         Demonstration of msg mail sys
N      7  487  6-Jun-1984 THE-MAIL-SYSTEM  Automatic receipt
N      8  321  6-Jun-1984 agazzi           Meeting about Support Documents
N      9  534  6-Jun-1984 census           Production Schedule
N     10  213  6-Jun-1984 census           Machine Arrangements

<-
```

## How to Enter MSG Commands

MSG commands are one letter abbreviations for the command names they represent. This means that you type one letter, and the system will complete the command name as well as the rest of the words in the command sentence for you.

For example, if you want to see a list of all the MSG commands you should enter a ? (question mark). MSG will complete the words in the command name and then print the list of commands:

```
<- ? MSG Help The following commands are recognized by MSG:
```

```
The following commands are recognized by MSG:
```

```
a[nswer]   message
b[ackup]   to previous message and type it
d[ele]te   message(s)
e[xit]     and update old file
.
.
.
```

Many MSG commands will require additional information from you before they can be completed. The **Type Command**, which is used to print messages on your screen, is a good example of this, since it asks you which messages you want to read.<sup>2</sup>

As you can see from the example below, if you enter the letter **t** MSG will complete the letters in the word "type" and then ask you which messages you want to read; it does this by printing the phrase "(message sequence);" in the example, we ask that message number 1 be printed:

```
<- type (message sequence) 1 (RET)
```

---

<sup>2</sup>The **Type Command** is described in the section entitled "Reading Messages" which begins on page 15.

## Getting Help

There are five different ways to get help from the mail system, each of which is illustrated here. Only part of what you will see if you actually type any of the Help commands is shown in the examples in order to save space in the manual.

(1) Getting a List of all MSG Commands:

If you want to see a list of all the MSG commands, their command names, and their meanings, type a ? (question mark) when the <- prompt appears at the left of your terminal screen.

An example of the ? Command, and a part of MSG's response looks like this:

```
<- ? MSG Help The following commands are recognized by MSG:
a[nswer]    message
b[ackup]    to previous message and type it
c[urrent]   message number and file
d[elete]    message(s)
e[xit]      and update old file
.
.
.
```

(2) Getting a Narrative Introduction to MSG:

If you type a % (percent sign) you will get an introduction to the mail system. The introduction gives you an overview of MSG, suggests commands that you might try, and makes recommendations on how to best use the system.

An example of the % Command, and a small part of what you will see if you type the command looks like this:

## Getting Help, Continued

<- % Introduction to MSG

If you are a new MSG user, you probably need ONLY the following commands:

- t type message(s) on terminal; common options are 'a' for all messages or '<n>' (where <n> is an integer) for message <n>.
- d delete a message after reading it; common options as above.
- e exit MSG.
- q quit MSG and forget any actions you took, or changes you made.  
New messages will still be marked new, even if you have read them;  
messages sending return receipts will send them again when you use MSG;  
deleted messages will still be there.

### (3) Getting a Description of the Changes to the MSG System:

If you type a # (pound sign) MSG will print a description of the changes that have been made to the system in the recent past.

An example of the # Command and a portion of what you will see on your screen when you use it, looks like this:

<- # MSG News

Mail System

16-Mar-84

Page 1

Several new features have been added to the mail system in the past couple of months. This document is meant to outline those features currently available until the normal documentation catches up.

#### I. New features of sndmsg

##### \* New commands to sndmsg

b[cc] - add addresses to Bcc field in header

c[c] - add addresses to cc field in header

## Getting Help, Continued

### (4) Finding Out What Options are Available for a MSG Command:

Many of the MSG commands will ask that you enter additional information. If you are unsure about what to enter, type a ? in response to the request, and MSG will print a list of all the options available to you at that point. For example, the **Type Command** will ask you for a "message sequence;" to find out what you should enter, type a ? in response to the prompt, as shown below:

<- **type** (message sequence) ? may be any of the following:

- (1) Any single message number, as listed in the headers.
- (2) Any two message numbers separated by ':' or '-'. This implies a range of messages (e.g. 2-5).
- (3) Any sequence of the previous two types, separated by commas. For example, '1,3,5-7,10'.

### (5) Getting a Narrative Description of an Individual MSG Command:

You can get a narrative description of any MSG command by typing an **I** followed by the name of the MSG command. To use the **I Command** to ask for information about the **Type Command**, follow the example shown below. Only part of the response you can expect from MSG is shown here.

<- **information** - type command character: **t**

Type (message sequence) <MSG-SEQUENCE>

This command displays the messages specified. If more than one message is specified, the user is prompted with "[type SPACE for next message]" after each message. In addition, if a particular message is larger than one screenful, the user is prompted after each screenful.

---

## Leaving MSG

---

There are two different ways to leave MSG. The effects of these two commands differ dramatically. Therefore, it is important that you read this section carefully.

Most of the commands you will read about in this manual change the "state" of your messages.

When, as you will see later, you read a "new" message, it loses its newness and becomes an "old" message; if you read a message that has a return receipt attached, MSG remembers that a return receipt has been sent. Likewise, MSG keeps track of which messages you've marked for deletion. These examples are only some of the types of record keeping that MSG does. Keep these facts in mind as you read on.

## Leaving MSG, Continued

### The Exit Command

The **Exit Command** is the recommended way to leave MSG. When you use this command, MSG will make all the changes you've made to your messages permanent.

This means:

- (1) Any messages that you've read will lose their "new" designation, and become "old" messages.
- (2) If you have read a message that had a return receipt attached to it, MSG will remember that it has sent the return receipt.
- (3) Any messages that you've marked for deletion will be removed from your mail box or folder.

To use the **Exit Command**, enter the letter **e**. MSG will ask that you type a space to confirm that you wish to "exit" from the system and update your incoming mail box. After a short pause while MSG makes the appropriate changes to your mail box or folder, you will be returned to DCL, and the **\$** prompt:

```
<- exit and update old file 'rix$user:[smith]mymail'
```

```
[type SPACE to confirm] (Space) updating...
```

```
$
```

## Leaving MSG, Continued

### The Quit Command

The **Quit** Command is the “back door” to the MSG system. It is only to be used when you want to “pretend” that you haven’t entered MSG at all. When you use the **Quit** Command, all the changes you’ve made to your messages will be discarded.

This means that:

- (1) MSG will forget that you’ve read messages. Any messages that were “new” when you started will retain their “new” designation, even if you *have* read them.
- (2) If you’ve read a message that had a return receipt attached, MSG will have sent a notification to the message’s sender saying that you’ve received and read the message.

*However*, if you leave with the **Quit** Command, MSG will forget that it’s sent the return receipt. Therefore, the next time you enter MSG and read the same message a second time, a *second* notification will be sent saying that you’ve read the message. This tends to irritate the sender.

- (3) MSG will ignore your delete commands; instead, your messages will remain in your incoming mail box or folder. If you were going to use the **Quit** Command to “keep” deleted messages, look at the section entitled “Recovering Deleted Messages” instead.

If you must use the **Quit** Command, enter the letter **q**. If you have made any changes, MSG will request that you type a space to confirm that you want to quit and throw away the changes:

```
<- quit
```

```
[Type SPACE to confirm] (Space)
```

```
$
```

## Leaving MSG, Continued

### The Overwrite Command

There may be times when you want the results of your work made permanent immediately. This can be done by using the **Overwrite Command**. All the changes that are listed in the section entitled "The Exit Command" will be made. You will normally use this command when you've deleted a lot of messages, and want them removed from your mail box or folder but are not ready to leave MSG.

Use the **Overwrite Command** by entering the letter **o**. MSG will ask that you confirm the command by typing a space (this will protect you from accidentally using the command when all you've done is made a typographical error).

MSG will automatically renumber and print the headers of the "new" messages that remain in your incoming mail box or folder. To see the headers of *all* the messages, refer to the section entitled "Headers, How to Tell What's in Your Mail Box."

In this example, the user has deleted several of his messages, and now wants to work with the remaining three:

```
<- Overwrite old file 'rix$user:[smith]mymail'
```

```
[type SPACE to confirm] (Space) updating...
```

```
3 messages/ 3 new
```

1	304	6-Jun-1984	virginia	maintenance
2	525	6-Jun-1984	schroeder	Using vtroff
3	481	6-Jun-1984	nadel	Meeting to discuss contracts

```
<-
```

---

## Reading Messages

---

You use the **Type Command** to read the messages that are in either your incoming mail box, or one of your file folders.<sup>3</sup> Use the **Type Command** to read your messages by entering the letter **t** when you see the **<-** prompt at the left side of your terminal screen. The **Type Command** will ask you which message or messages you want to read.

Before showing an example, there are two terms which MSG uses that must be explained, namely "message sequence" and "message header."

### Message Sequences

"Message Sequence" is a term which MSG uses to ask you which of the messages in your mail box you want to read, delete, forward, or answer, etc.

In all, there are twelve different ways of specifying a message sequence,<sup>4</sup> however, the four shown here will take care of most of your needs. You can tell MSG which messages you want to refer to by giving it:

- (1) A message number: this can be a single number, a series of numbers, a range of numbers, or a combination of all three.
- (2) The name of the sender;
- (3) The subject of the message;
- (4) All messages.

Whenever MSG asks you for a "message sequence," you must give it a list which contains one of these four items.

The information you will use to create this list is kept in the "message header;" message headers are explained on the next page.

---

<sup>3</sup> Folders are described in the section entitled "A Filing System for Messages," which begins on page 35.

<sup>4</sup> See the Command summary for the remaining eight message sequences.

## Reading Messages, Continued

### Message Headers

As you will remember, when the MSG system starts, it introduces itself by printing a banner and then prints a list showing all the new messages in your mail box:

```
$ msg (RET)

Software Tools MSG System
type ? for help
type # for news
type % for intro

Many new features (3/16/84) - see news!

10 messages / 10 new
N      1  427  6-Jun-1984 virginia      vacation schedules
N      2  304  6-Jun-1984 virginia      maintenance
N      3  525  6-Jun-1984 schroeder     Using vtroff
N      4  347  6-Jun-1984 census       test
N      5  481  6-Jun-1984 nadel        Meeting to discuss contracts
NR     6  546  6-Jun-1984 mccarthy     Demonstration of msg mail sys
N      7  487  6-Jun-1984 THE-MAIL-SYSTEM Automatic receipt
N      8  321  6-Jun-1984 agazzi       Meeting about Support Documents
N      9  534  6-Jun-1984 census       Production Schedule
N     10  213  6-Jun-1984 census       Machine Arrangements

<-
```

Each line in the list is called a "message header," and contains six columns of information about the individual messages. These columns contain:

- (1) A message "status" which tells you whether a message is new (N), deleted (D), answered (A), or if the sender has asked to be notified at the time you read the message (R).

In the example above, all the messages are new, which is indicated by an **N** in the first column of the message header. Also, the sender of message 6 has asked that he be notified when you've read the message. This is called a "return receipt" and is indicated as an **R** in the message status column.

## Reading Messages, Continued

- (2) The message number. Each message is assigned a number when the system puts it in your mail box.
- (3) The size of the message in characters. Normally, the size of a message will not be of any interest to you.
- (4) The date the message was mailed. All the example messages were sent on June 6, 1984.
- (5) The name of the person who sent the message.
- (6) The subject of the message.

The examples you will see on the next several pages will show you how to read your mail using the four basic message sequences described here.

## Specifying a Message by Number

To use a number to tell MSG which message you want to read, use the **Type Command** by entering the letter **t** and supplying the appropriate message number when MSG asks you for a "message sequence."

There are four ways you can specify a numeric message sequence: 1) by entering a single number; 2) by entering a series of numbers; 3) by entering a range of numbers; and 4) by entering a combination of all three. The four examples on the next pages show you how to specify these numeric message sequences.

### (1) A Single Number:

In this first example, we will read the second message in the mail box. If you look back at the example of the message headers on page 16, you will see that the second message is from "virginia" and is about "maintenance."

To read this message, first use the **Type Command** by entering the letter **t**. Then, when MSG asks you for a message sequence, enter the number **2**, just as you see in the example below:

```
<- type (message sequence) 2 (RET)
```

```
(message 2, 289 characters)
```

```
Date: Wed, 6 Jun 84 09:56:15 PDT
```

```
From: virginia (lblh csam sventek) @ lbl-h.arpa
```

```
Message-Id: <840606095610.006@lbl-h.arpa>
```

```
Subject: maintenance
```

```
To: smith
```

```
Please contact me about maintenance agreements for the  
computers.
```

```
Virginia
```

```
<-
```

## Specifying a Message by Number, Continued

### (2) A Series of Numbers:

This example shows you how to use the **Type Command** to read a series of messages. When you enter a series of numbers, you separate each number with a comma, as shown below.

Whenever you print more than one message at a time, MSG will pause after printing each message to ask you if you want to see the next one; it does this by printing the phrase “[type SPACE for next message].” If you want to see the next message, hit the space bar once; if you do not, enter a **q** and you will be returned to the **<-** prompt.

```
<- type (message sequence) 1,2 (RET)
```

```
(message 1, 412 characters)
```

```
Date: Wed, 6 Jun 84 09:54:38 PDT
From: virginia (lbh csam sventek) @ lbl-h.arpa
Message-Id: <840606095433.005@lbl-h.arpa>
Subject: vacation schedules
To: smith
cc: quong
```

```
Sirs:
```

```
Vacation schedules for your staff have been posted outside
the main office on the third floor.
```

```
If you have questions, please call me on extension 5216. Virginia
```

```
[type SPACE for next message] (Space)
```

```
(message 2, 289 characters)
```

```
Date: Wed, 6 Jun 84 09:56:15 PDT
From: virginia (lbh csam sventek) @ lbl-h.arpa
Message-Id: <840606095610.006@lbl-h.arpa>
Subject: maintenance
To: smith
```

```
Please contact me about maintenance agreements for the
computers. Virginia
```

## Specifying a Message by Number, Continued

### (3) A Range of Numbers:

In this example, the **Type Command** is used to print a range of messages. You enter a range of numbers by placing a dash between the first and last number of the messages you want printed. As in the previous example, if you print more than one message at a time, MSG will ask you if you want to see the next message. (Only message 1 is shown in the example.)

```
<- type (message sequence) 1-3 (RET)
```

```
(message 1, 412 characters)
```

```
Date: Wed, 6 Jun 84 09:54:38 PDT
```

```
From: virginia (lbh csam sventek) @ lbi-h.arpa
```

```
Message-Id: <840606095433.005@lbi-h.arpa>
```

```
Subject: vacation schedules
```

```
To: smith
```

```
cc: quong
```

```
Sirs:
```

```
Vacation schedules for your staff have been posted outside  
the main office on the third floor.
```

```
If you have questions, please call me on extension 5216. Virginia
```

### (4) Combining Numeric Message Sequences:

In this example, the three numeric message sequences are combined. The **Type Command** is used to print a series of messages (messages 1 and 3) a single message (message 6), and a range of messages (8 through 10). In order to save space, we will show only the command, and not the messages themselves.

```
<- type (message sequence) 1,3,6,8-10 (RET)
```

```
<-
```

---

## **Specifying a Message by Sender Name**

---

You can tell MSG that you want to read the messages sent to you by a particular person by using the "from" message sequence.

If you glance back at the example message headers on page 16 you will see that the name of the sender is listed in column five of each header.

To print messages from a particular person, use the **Type Command** by entering the letter **t** and specify the "from" message sequence by typing the letter **f**.

When you use this message sequence, MSG will ask you for the name of the person by prompting you with the phrase "from string:" Type in the person's name followed by a carriage return. MSG will print out *all* the messages from that person.

In the example on the next page, the messages sent by a user of the "census" account are requested:

## Specifying a Message by Sender Name, Continued

<- type (message sequence) from string: census (RET)

(message 4, 347 characters)  
Date: Wed, 6 Jun 84 10:07:36 PDT  
From: census (lblh csam) @ lbl-h.arpa  
Message-Id: <840606100736.009@lbl-h.arpa>  
Subject: test  
To: smith

We are currently running tests of the production software.

[Type SPACE for next message] (Space)

(message 9, 534 characters)  
Date: Wed, 6 Jun 84 10:07:36 PDT  
From: census (lblh csam) @ lbl-h.arpa  
Message-Id: <840606100736.009@lbl-h.arpa>  
Subject: Production Schedule  
To: smith

We will be creating a production schedule during the next week. Let me know if you have any comments.

[Type SPACE for next message] (Space)

(message 10, 347 characters)  
Date: Wed, 6 Jun 84 10:07:36 PDT  
From: census (lblh csam) @ lbl-h.arpa  
Message-Id: <840606100736.009@lbl-h.arpa>  
Subject: Machine Arrangements  
To: smith

Do you have any comments about the machine move next week?

<-

## Specifying a Message by Subject

Each message in your mail box contains a subject field that tells you what the message is about. For example, if you glance back at the example headers on page 16 you'll see that the subject fields for two messages contain the phrase "meeting." If you want to refer to these messages without knowing the message numbers or sender names, you can use the "Subject" message sequence.

Use the **Type Command** by entering the letter **t** and specify the "subject" message sequence by typing the letter **s** when MSG asks you for a message sequence. You will be asked for the word or words you want MSG to search for in the subject field. You can enter any part, or all, of the words you see in the subject field; in this example we ask to see all the messages whose subject field contains the word "meeting."

MSG looks at the subject field on each message in your mail box trying to find an *exact* match of the word, words, or part of a word you type in.

<- type (message sequence) Subject string: **meeting** **RET**

(message 5, 481 characters)

Date: Wed, 6 Jun 84 10:07:36 PDT  
From: nadel (lblh csam lesta) @ lbl-h.arpa  
Message-Id: <840606100736.009@lbl-h.arpa>  
Subject: Meeting to discuss contract mods  
To: smith

Now that I am back from vacation, I think we should meet  
to discuss contracts.  
Is Tuesday afternoon at 2:00 ok on your schedule?

[Type SPACE for next message] **(Space)**

(message 8, 481 characters)

Date: Wed, 6 Jun 84 10:07:36 PDT  
From: agazzi (lblh csam carole) @ lbl-h.arpa  
Message-Id: <840606100736.009@lbl-h.arpa>  
Subject: Meeting about Support Documents  
To: smith

Our Support Document meeting is set for Monday at 2.

<-

## Reading All Messages

If you only have a few messages in your mail box, or some free time and a lot of messages, you can ask MSG to print them all at once.

This is done by specifying the message sequence "all." Use the **Type Command** and enter the letter **a** when MSG asks for a message sequence. The letter **a** is an abbreviation for the word "all."

Your messages will be printed one after another with a pause between them while MSG asks if you want to see the next message.

If you want to see the next message, hit the space bar once; if you don't, enter a **q** and you will be returned to the **<-** prompt.

An example of the command is shown below, but to save space we won't show the messages.

```
<- type (message sequence) all
.
.
.
<-
```

---

## How to Tell What's in Your Mail Box

---

There may be times when you are confused about the contents of your mail box or folder; for example, you may have just read a rather lengthy message and not be able to remember which message you wanted to read next. Or, as more frequently happens, you may have old messages in your mail box whose message sequences do not appear automatically when you start the MSG system. Since almost every MSG command requires that you specify a message sequence, it is important for you to have this information handy. Therefore, MSG has provided you with a **Header Command** which enables you to examine the headers of the messages in your mail box or folder, so you can quickly check message numbers, subjects, or sender names.

MSG will ask that you indicate which message headers you want listed by asking you for a message sequence. You may enter any of the four basic message sequences described earlier. The examples in this section show each message sequence in detail to help you reinforce the things you've already learned about message sequences.

### (1) Looking at the Message Headers on All Messages

To look at the headers on all the messages in your mail box, use the **Header Command** by typing the letter **h**; when MSG asks you for a message sequence, specify "all" by typing the letter **a**.

```
<- headers (message sequence) all

N      1   427  6-Jun-1984 virginia      vacation schedules
N      2   304  6-Jun-1984 virginia      maintenance
N      3   525  6-Jun-1984 schroeder     Using vtroff
N      4   347  6-Jun-1984 census        test
N      5   481  6-Jun-1984 nadel         Meeting to discuss contracts
NR     6   546  6-Jun-1984 mccarthy      Demonstration of msg mail sys
N      7   487  6-Jun-1984 THE-MAIL-SYSTEM Automatic receipt
N      8   321  6-Jun-1984 agazzi        Meeting about Support Documents
N      9   534  6-Jun-1984 census        Production Schedule
N     10   213  6-Jun-1984 census        Machine Arrangements

<-
```

## How to Tell What's in Your Mail Box, Continued

### (2) Looking at Headers by Number:

If you want to look at the header on a single message, use the **Header Command** by typing an **h**. When MSG asks you for a message sequence, enter a single number, as shown below:

```
<- headers (message sequence) 1 (RET)
  1  427  6-Jun-1984 virginia  vacation schedules
<-
```

To look at the headers on a series of messages, use the **Header Command** by typing an **h**, and when MSG asks you for a message sequence, enter the message numbers separated by commas:

```
<- headers (message sequence) 1,3,5 (RET)
  1  427  6-Jun-1984 virginia  vacation schedules
  3  525  6-Jun-1984 schroeder  Using vtroff
  5  481  6-Jun-1984 nadel     Meeting to discuss contracts
<-
```

To look at a range of message headers, use the **Header Command** by typing an **h**; when MSG asks you for a message sequence, enter the first and last message numbers of the range (separated by a dash).

```
<- headers (message sequence) 1-3 (RET)
  1  427  6-Jun-1984 virginia  vacation schedules
  2  304  6-Jun-1984 virginia  maintenance
  3  525  6-Jun-1984 schroeder  Using vtroff
```

## How to Tell What's in Your Mail Box, Continued

### (3) Looking at Message Headers by Referring to Sender Name:

If you want to look at the headers on the messages sent to you by a specific person, use the "from" message sequence. To do this, use the **Header Command** by typing the letter **h**; when MSG asks you for a message sequence enter the letter **f** (an abbreviation for the word "from"). MSG will then ask you to enter the sender's name:

```
<- headers (message sequence) from string: census (RET)
N   4   347   6-Jun-1984 census   test
N   9   534   6-Jun-1984 census   Production Schedule
N  10   213   6-Jun-1984 census   Machine Arrangements
<-
```

### (4) Looking at Messages by Referring to Their Subject:

If you want to see all the message headers on messages regarding a particular subject, use the "subject" message sequence. Use the **Header Command** by typing the letter **h**; when MSG asks you for a message sequence, enter the letter **s**. MSG will then ask you for a word it is to look for in the subject fields of all your messages.

```
<- headers (message sequence) subject string: meeting (RET)
N   5   481   6-Jun-1984 nadel   Meeting to discuss contracts
N   8   321   6-Jun-1984 agazzi  Meeting about Support Documents
<-
```



---

## Deleting Messages

---

Your mail messages will stay in your mail box until you delete them or move them elsewhere, and while you will want to keep many of your messages for future reference, you will want to throw away a large portion of them. (Electronic mail accumulates as quickly as paper mail.)

The MSG system makes it easy to file messages, and even easier to throw them away. This section shows you how to delete your messages; the section entitled "A Filing System for Messages" will show you how to file them.

There are two steps to deleting messages: 1) use the **Delete Command** by entering the letter **d**; 2) specify a message sequence, telling MSG which messages to delete.

The messages you ask to have deleted will be removed from your incoming mail box *if* you leave the message system by using the **Exit** or **Overwrite Commands**. If you leave the system any other way, (by using the **Quit Command**, or accidentally turning off your terminal while reading your mail) the messages will remain in your incoming mail box.

A special section entitled "Leaving MSG" summarizes the different effects of the **Exit** and **Quit Commands**.

The examples in this section show you how to delete messages using the four basic message sequences shown in the previous section.<sup>5</sup>

---

<sup>5</sup>The four basic message sequences are described in more detail in the section entitled "Reading Messages."

## Deleting Messages, Continued

### (1) Deleting a Message by Number:

To delete messages by number, use the **Delete Command** by entering the letter **d**. When MSG asks you for a message sequence, enter one of the four numeric message sequences: a single number, a series or range of numbers, or all three.

The first example shows you how to delete a single message; the second shows you how to delete a series of messages; the third deletes a range of messages; and the fourth example shows you how to combine the numeric message sequences. Remember, these messages will not be deleted unless you leave the MSG system by using the **Exit** or **Overwrite Commands**.

```
<- delete (message sequence) 1 (RET)
<- delete (message sequence) 2,4 (RET)
<- delete (message sequence) 6-7 (RET)
<- delete (message sequence) 1,2,4,6-7 (RET)
<-
```

### (2) Deleting Messages by Sender Name:

To delete all the messages sent to you by a particular person, use the "from" message sequence. To do this, use the **Delete Command** by entering the letter **d**. When MSG asks you for a message sequence, enter the letter **f** which tells the system that you want to delete messages "from" a specific user. MSG will then ask you for the user's name. Again, these messages will not be deleted unless you use the **Exit** or **Overwrite Commands** to leave MSG. In the example below, the messages sent by a user named "Census" are deleted:

```
<- delete (message sequence) from string: census (RET)
<-
```

## Deleting Messages, Continued

### (3) Deleting Messages by Subject:

To delete messages by referring to their content, use the "subject" message sequence. As you can see from the example, you must enter the letter **s** when MSG asks you for a message sequence (**s** is an abbreviation for "subject"). MSG will ask you for the word or words it is to look for in the subject field in order to determine which messages it is to delete.

In the example, all the messages with the word "meeting" in their subject field will be deleted (if you use the **O**verwrite Command or leave MSG by using the **E**xit Command).

```
<- delete (message sequence) subject string: meeting (RET)
<-
```

### (4) Deleting All Messages:

You can delete all your messages at one time by using the message sequence, "all." Be careful, though, you don't want to throw away something you want to keep.

```
<- delete (message sequence) all messages (RET)
```

If you really want all your messages deleted, be sure to leave the MSG system by using the **E**xit Command, or use the **O**verwrite Command.



---

## Recovering Deleted Messages

---

Every once in a while you will find that you've deleted the wrong message, or that you really did want to keep the message you just deleted. MSG allows you to "undelete" a message or set of messages, as long as you haven't left the MSG system. Use the **Undelete Command** by entering the letter **u**; it will ask you which of the deleted messages you want to recover by prompting you for a "message sequence."

The examples below show you how to recover deleted messages using the four basic message sequences: number, sender name, subject, and all messages.

- (1) Recovering Messages by Number:

```
<- undelete (message sequence) 1 (RET)
<-
```

- (2) Recovering Messages by Sender Name:

```
<- undelete (message sequence) from string: census (RET)
<-
```

- (3) Recovering Messages by Subject:

```
<- undelete (message sequence) subject string: schedule (RET)
<-
```

- (4) Recovering All Deleted Messages:

```
<- undelete (message sequence) all messages (RET)
<-
```



---

## A Filing System for Messages

---

Every correspondence system you've ever used has a filing system of some sort; it might be a filing cabinet with carefully labeled drawers and neatly ordered manila folders, or a corner of your desk which you've designated as your filing area.

Whatever your filing style, MSG can help you to bring order to your correspondence by providing facilities which enable you to create filing "folders" for your messages.

If you are able to categorize the kinds of mail you send and receive, using folders to file your messages in an order that makes sense to you can greatly increase your efficiency and make your correspondence tasks easier to deal with in general.

The person we've been using as an example throughout the manual sends and receives mail that relates to the three major functions of his job: administration, software support, and project coordination. In order to keep track of his mail, he has created a folder for each of these functions.

As he reads his mail, he decides which folder each message should be filed in. For example, whenever he receives a piece of mail that has something to do with administration, such as a copy of his people's vacation schedules, he files it in his "administration" folder. He files meeting announcements and agendas in his "project" folder.

When people send him software questions through the MSG system, he files both their question and his answer in the "software" folder. If, at a later time, he receives a message that contains the same software question, he can pull his previous response out of the "software" folder and forward it to the user (forwarding is explained later) rather than creating a brand new response.

This section describes 1) how to create folders; 2) how to put your messages into folders; and 3) how to access the messages once they are in the folder.

---

## Creating File Folders

The process of creating MSG folders is quite simple. For each MSG folder you want to create, you must supply:

**A Name:** Folder names can be of any length, however, there are two rules to follow when assigning a name to a folder:

- (1) Folder names must be preceded by a plus sign (+);
- (2) Folder names cannot include blanks; separate the words in the name with an underscore, for example, "+personnel\_actions"

**A Location:** Each MSG folder is stored in a VMS file which, for your convenience, should be kept in your main directory.

The most efficient method is to provide MSG with a full file specification so that you can use the folders whether or not you are in your main directory at the time you read your mail.<sup>6</sup>

**A Description:** The description should be a short phrase that will tell you what kind of messages the folder contains, just in case you've forgotten.

MSG will create the folders automatically. The example on the next page shows you exactly what to type in order for MSG to create the folders.

---

<sup>6</sup> If you are unfamiliar with VMS files, please take the time to look at the "Introduction to DCN;" it contains a complete description of how to name and manipulate files. Alternatively, your system manager can help you.

However, briefly stated, a full file specification includes a **disk name**, a **[directory]** name, a **file name** (up to nine characters in length), and the three letter extension, **msg**.

## Creating File Folders, Continued

The definitions of your MSG folders must be kept in a VMS file named "folders." in your main directory. You can use an editor to create the "folders." file, but the **create** command itself may be easier for new users who are unfamiliar with an editor.

As shown in the example below, you type the command "**create folders.**" when you see the \$ sign prompt from DCL. ("**Folders.**" is a special name, which MSG has been programmed to look for. Therefore, you must be sure to include the "." in the file name.) After entering the command, type in a line describing each folder by entering a name, location, and description for each folder. The **create** command will not prompt you for anything, so you can start typing right away.

Of course you should replace the folder names, locations, and descriptions in the example with your own folder names, their location, and a description that is meaningful to you.

After you have described all your folders, end the **create** command by holding down the button labeled "CTRL" and, at the same time, typing a **z** (you will see the characters "**^Z**" on your screen).<sup>7</sup>

After ending the **create** Command you will be returned to the DCL prompt, \$, and the folders will be ready to use. Since each folder is created at the time the first message is stored into it, a listing of your directory will not show any of the files names you've just defined.

```
$ create folders. (RET)

+admin      rix$user:[smith]admin.msg      Administrative issues (RET)
+project    rix$user:[smith]project.msg  Notes and agendas for projects (RET)
+software   rix$user:[smith]software.msg  Help with software problems (RET)

^Z

$
```

<sup>7</sup> Use the CTRL (control) key the same way you would use the shift key, that is, hold the CTRL key down until you've typed the z and then release the CTRL key.

---

## Moving Messages Into Folders

---

The **Move Command** is used to transfer messages into a **MSG** folder. (It can also be used to move messages out of the **MSG** system and into a **VMS** file.<sup>8</sup>) At the same time, the messages you transfer will be marked as deleted in your incoming mail box, which means that the original copy of the message will be thrown away if you use the **Overwrite Command** or the **Exit Command** to leave **MSG**.<sup>9</sup>

The **Move Command** will ask you two things: 1) which message you want to move, and 2) which folder or file you want the message stored in.

In order to illustrate the **Move Command**, we will assume that the incoming mail box contains the messages you see below:

N	1	427	6-Jun-1984 virginia	vacation schedules
N	2	304	6-Jun-1984 virginia	maintenance
N	3	525	6-Jun-1984 schroeder	Using vtroff
N	4	347	6-Jun-1984 census	test
N	5	481	6-Jun-1984 nadel	Meeting to discuss contracts
NR	6	546	6-Jun-1984 mccarthy	Demonstration of msg mail sys
N	7	487	6-Jun-1984 THE-MAIL-SYSTEM	Automatic receipt
N	8	321	6-Jun-1984 agazzi	Meeting about Support Documents
N	9	534	6-Jun-1984 census	Production Schedule
N	10	213	6-Jun-1984 census	Machine Arrangements

---

<sup>8</sup> Moving messages to **VMS** files is described in the section entitled "Moving Your Messages to a File."

<sup>9</sup> Refer back to the section entitled "Deleting Messages" for more information.

## Moving Messages Into Folders, Continued

The **Move Command** is used by typing an **m**. **MSG** will ask you for a list of the messages you wish to move, and the name of the file you wish to move them into.

In this section, the messages are to be transferred into **MSG** folders. Since the **Move Command** accepts **VMS** file names as well as folder names as places into which you can move a message, you must be careful to include the **+** (plus sign) as part of any folder name.

- (1) The first message in the incoming mail box contains a list of vacation schedules, and should be filed in the "+admin" folder. We will specify this message by using the "subject" message sequence:

```
<- move (message sequence) Subject string: vacation (RET)
into file name: +admin (RET)
<-
```

- (2) To move the message from the user named "schroeder" into the "+project" folder, use the "from" message sequence, as shown below:

```
<- move (message sequence) from string: schroeder (RET)
into file name: +software (RET)
<-
```

## Moving Messages Into Folders, Continued

- (3) Messages 2 and 5 should both be filed in the "+project" folder, and are specified with by a numeric message sequence.

```
<- move (message sequence) 2,5 (RET)
```

```
into file name: +project (RET)
```

```
<-
```

## Accessing Messages Stored in Folders

Thus far you have been shown the commands that enable you to create folders and, once the folders have been made, move messages into them. In this section, you will be shown the commands that let you work with the messages that have been stored in the folders.

In order to be flexible enough to satisfy your needs, MSG has provided you with two different ways of telling it which folder to use.

The first allows you to move between folders (or your incoming mail box) from within MSG itself. The second allows you to have a folder read in at the time the MSG session is started.

Each of these methods has its own advantage: Accessing folders from within MSG, at the <- prompt, is perhaps the most common since it is so easy to switch back and forth between folders during a single MSG session. On the other hand, if you know ahead of time that you want to work with the messages in a particular folder, specifying that MSG begin your session by reading in that folder is faster.

Both of these methods are described in this section.

## Accessing Messages Stored in Folders, Continued

### Reading Messages From Within MSG

The command used to access the contents of a folder from within MSG is the **Read Command**. **Read** will ask you to give it a folder name.

What happens next will depend on what you were doing before you issued the **Read Command**.

- (1) If you have been working with messages in your incoming mail box (or another folder), MSG will want to know if you want the effects of your work recorded. (For example, if you have been deleting messages or reading new messages, you will probably want MSG to do the deletions before moving over to work with another folder. By the same token, you will most likely want MSG to record the fact that the messages you've just read are no longer "new.")
- (2) If you have *not* been working with other folders (or your incoming mail box), MSG will immediately access the next folder.

In the examples on the next page, the user has been reading and deleting messages in his incoming mail box and now wants to move over to the "**+project**" folder. He would like MSG to make note of the fact he's read some "new" messages and deleted others, and requests that MSG update his incoming mail box, "mymail."

These examples are rather unusual in that MSG prints several new "informative phrases," so we will describe them line by line:

- (1) the user issues the **Read Command** by typing the letter **r**.
- (2) MSG asks for a file name, and the user responds by entering the folder name "**+project**."
- (3) MSG asks if it should update the incoming mail box, "mymail." (In the first example, the user wants to update the incoming mail box and enters a **(Space)**; the second example shows you what happens if you do not want to record changes to the current folder or incoming mail box.)
- (4) MSG prints a message that the incoming mail box, "mymail," is being "updated."
- (5) Finally, MSG prints a message saying that it is "reading" in the "**+project**" folder.

## Accessing Messages Stored in Folders, Continued

To use the Read Command, enter the letter **r**. MSG will then ask you for a folder name, and whether you want your previous work recorded:

```
<- read file name: +project (RET)

Update file 'rix$user:[smith]mymail' first?

[type SPACE to confirm] (Space) updating...reading...

2 messages / 2 new
  1  304  6-Jun-1984 virginia  maintenance
  2  481  6-Jun-1984 nadel    Meeting to discuss contracts

<-
```

In the next example, the user wants to transfer to the “+project” folder, but *does not* want the work he’s done in the incoming mail box recorded. So, when MSG asks him if he wants to update the “mymail” file, he enters an **n**. (He could have entered any character other than a **(Space)** and it would have had the same effect; **n** — short for “no” just seems to make more sense.)

```
<- read file name: +project (RET)

Update file 'rix$user:[smith]mymail' first?

[type SPACE to confirm] n reading...

2 messages / 2 new
  1  304  6-Jun-1984 virginia  maintenance
  2  481  6-Jun-1984 nadel    Meeting to discuss contracts

<-
```

## Accessing Messages Stored in Folders, Continued

### Starting a MSG Session With a Folder

If you know in advance that you want to work with messages that you've stored in a folder, you can request that your MSG session begin with that folder.

When you do this, MSG will skip the step that reads in your incoming mail box and will go directly to reading the messages in the folder.

The first thing MSG does when you start it, is read in the messages in your incoming mail box. In order to prevent this from happening, and redirect MSG to the appropriate folder instead, you must signal MSG as early as possible.

This is done by putting the folder name directly after the **msg** Command (don't forget to include the plus sign as part of the folder name).

```
$ msg +project (RET)
```

```
Software Tools MSG System
```

```
type ? for help
```

```
type # for news
```

```
type % for intro
```

```
Many new features (3/16/84) - see news!
```

```
2 messages / 2 new
```

```
1 304 6-Jun-1984 virginia maintenance
2 481 6-Jun-1984 nadel Meeting to discuss contracts
```

```
<-
```

## Viewing and Adding Folders

MSG gives you a command to use whenever you've forgotten your folder names, where they are located, or what kinds of messages they contain. The command name is **View**; it will prompt you for the names of the folders whose definitions you wish to see.

Your options are to tell it that you want to see the names and definitions for 1) all folders, or 2) a specific folder.

**All Folders** To see the names and definitions of **all** the folders created in the previous example, use the **View Command** by entering the letter **v**; when MSG prompts for a folder name, enter the letter **a** which in this case stands for the word "all."

```
<- View folder all
+admin      rix$user:[smith]admin.msg      Administrative issues
+project    rix$user:[smith]project.msg  Notes and agendas for projects
+software   rix$user:[smith]software.msg  Help with software problems
<-
```

**A Specific Folder** To view the definition of a specific folder, use the **View Command** and enter the folder name when MSG prompts you, being sure to include the plus sign (+) as part of the folder name.

```
<- View folder +software (RET)
+software   rix$user:[smith]software.msg  Help with software problems
<-
```

## Viewing and Adding Folders, Continued

You can create a new folder at any time, by editing the “**folders.**” file and adding the name, location, and definition of the new folder.

If you have used an editor before, adding a folder definition is easy; if you haven't, you should probably ask someone to help you. Your local system manager is a good candidate.

The example below uses the editor EDT to add the definition of a folder in which to store “urgent” messages. Once you have edited the “**folders.**” file, the new folder, “urgent” will be ready to use.

```
$ edit folders. (RET)

1 +admin      rix$user:[smith]admin.msg      Administrative issues

*c (RET)

+admin      rix$user:[smith]admin.msg      Administrative issues
+project    rix$user:[smith]project.msg    Notes and agendas for projects
+software   rix$user:[smith]software.msg    Help with software problems
+urgent     rix$user:[smith]urgent.msg    Urgent Messages (RET)

^z
*exit (RET)
RIX$USER:[SMITH]FOLDERS.;2 4 LINES

$
```

---

## **Sending Messages**

---

Composing and sending messages is the responsibility of a portion of MSG called "sndmsg." Sndmsg is an efficient and flexible composition tool that you use whenever you want to write, forward, or answer a letter. Among the facilities it provides are carbon copies, return receipts, enclosures, and attachments.

There are, basically, four steps to sending a message: 1) filling out the header — the message's envelope, 2) attaching a return receipt request — this is optional, 3) writing the message, and 4) sending it.

Each of these steps and their facilities are described in this section with each variation on the theme given its own example.

---

## Filling Out the Header — the Envelope

---

Sndmsg uses a “fill in the blanks” approach to entering the information for the message header. If you will remember, a “header” contains basically the same information you would find on a paper envelope but its difference lies in the additional pieces of information you can include.

All counted, you will be asked to enter five different parts of the header:

**To:** In this field you enter the names of the people to whom you are sending the message. You can include as many names as you wish, separating the names with commas.

If you are sending mail to people who have their accounts on a machine other than your own, you must include their machine name along side their own name. To make it possible for MSG to differentiate between a user's name and a machine name, separate the two with an at sign (@). For example, to send a message to a user named “jones” whose account is on the “rx” machine, you will need to address the message to “jones@rx.”

If your message is to be delivered to people on your own machine as well as other machines, separate their addresses with commas: “virginia,jones@rx,schroeder.”

**Cc:** Carbon copies of your message will be sent to the people you list in this field. Their addresses should be entered in the same way you enter addresses in the “To:” field. <sup>10</sup>

**Bcc:** This field is used to send a covert copy of the message to the people you name. Their names will not appear on the copies of the message sent to others.

Since the “To,” “Cc,” and “Bcc” fields allow you to enter as many names as you wish, MSG will continue to prompt you for additional names until you enter a **RET** in response to the “To,” “Cc,” and “Bcc” prompts, as shown in the next example.

---

<sup>10</sup> “Cc” is beginning to be referred to as a “courtesy copy” since there's very little carbon in an electronic message, but it isn't used enough to keep it from sounding pretentious.

## Filling Out the Header - the Envelope, Continued

**Subject:** You should enter a description of your message in this field. Keep it short though, under 30 characters, or it won't fit on the message header.

**Return Receipt:** When you have finished filling out the preceding fields, MSG will ask you if you want to receive a return receipt when the addressee reads the message. You can answer yes if you do, and no if you don't. <sup>11</sup>

---

<sup>11</sup> Return Receipts are described in a separate section later in this manual.

## Filling Out the Header - the Envelope, Continued

The command used to “start” the Sndmsg portion of MSG is the Send Command. Use it by entering the letter **s**.

MSG will ask you for confirmation; this keeps you from ending up in Sndmsg when you’ve accidentally typed the wrong letter.

As soon as you enter the **(Space)**, Sndmsg will ask you to start filling out the header fields.

In this example, a message is sent to a person named “agazzi” on the user’s machine. Copies are sent to three people: “virginia” on the local machine, and “postle” and “jones” both of whom have accounts on other machines. In addition, the user wants to send “pasek” a copy of the message without letting the others know.

The subject of the message is “manuals,” and the user does *not* want a return receipt when the message is read.

One thing you should notice is that after the header has been completed, the prompt changes to **sm>** to show you that MSG has turned control over to Sndmsg. Also, Sndmsg tells you that you can type an **h** to get help. Sndmsg never completes a word, e.g., type, in the way that MSG does, and always requires a **(RET)** at the end before it does anything.

```
<- sndmsg [type SPACE to to confirm] (Space)
To: agazzi (RET)
To: (RET)
Cc: virginia,postle@rix,jones@rx (RET)
Cc: (RET)
Bcc: pasek@etadc (RET)
Bcc: (RET)
Subject: manuals (RET)
Do you want a return receipt? [y/n] n (RET)
If you need help, type h
sm>
```

---

## Leaving Sndmsg - When You Do not Want to Send the Message

---

If you change your mind about sending a message, you can use the Sndmsg **Quit** Command. This command acts differently than the MSG **Quit** Command in that it simply asks you if you really want to throw away the message that you're creating, and then returns you to the MSG prompt, <-.

To use the Sndmsg command, **Quit**, enter the letter **q** any time you see the sm> prompt. Sndmsg will ask that you type a **y** to confirm that you want to discard the message, or an **n** if you decide to keep the message.

For example, if the user wanted to discard this message, he would type:

```
sm> quit (RET)
Are you sure? [y/n] y (RET)
<-
```

## Changing the Header

There may come a time when you want to change the entries you've made to a message header. It's not unusual to want to add people to the "To," "Cc," or "Bcc" lists. On the other hand, you may find that the subject entry you've made is not representative of the message, or that the message is important enough to warrant a return receipt after all. MSG provides you with tools to make these changes and additions to the header.

Changes to the header fields are done by *adding* addresses to the "To," "Cc," and "Bcc" fields, or *redefining* the "Subject" and "Return Receipt" fields.

The Sndmsg Commands you see listed below can be used whenever you see the Sndmsg prompt, sm>.

t Prompts you for additional names for the "To" field.

c Will prompt you for additions to the "Cc" field.

b This command will let you add addresses to the "Bcc" field.

You can add as many names as you like, and the entries should be made in the same way they were when the header was originally filled out.

d s These letters are command abbreviations for "define subject." You should type the new subject entry on the same line as the "d s" command.

d r These are abbreviations for "define return receipt" command. With it, you can change your mind about whether or not you wish to receive a return receipt when the message is read.

## Changing the Header, Continued

In the following example, a copy of the message is sent to the user named "nadel" on the user's own local machine. This makes the fourth entry in the carbon copy field. Enter the letter **c** when you see the **sm>** prompt, and MSG will ask you for additional names to be placed on the "Cc" field:

```
sm> c (RET)
cc: nadel (RET)
cc: (RET)
sm>
```

Similarly, to add a name to the "To" field, enter the letter **t**:

```
sm> t (RET)
To: white (RET)
To: (RET)
sm>
```

## Changing the Header, Continued

The entry you have made to the "Subject" field of your message can be changed at any time prior to sending the message. This is done by using the **Define Command** and specifying the **Subject** field.

To change the subject field, enter the letters **d** and **s** (separated by a space). Type the new subject entry on the same line. In this example, the Subject field is changed from simply "Manuals" to "We have received the manuals."

```
sm> d s We have received the manuals (RET)
sm>
```

You can also use the **Define Command** to change the Return Receipt status by entering the letters **d** and **r** (separated by a space).

If you have *not* requested a return receipt, entering a **y** will reset the return receipt status to "yes;" if you *have* requested a return receipt, entering a **n** will reset the return receipt status to "no."

In an earlier example, the Return Receipt status for this message was set to "no." In this example, it is reset to "yes:"

```
sm> d r y (RET)
sm>
```

## Looking at the Header

MSG provides you with a set of commands to use if you ever become confused about who you're sending the message to or what you've said the message was about.

The **List Command** allows you to look at the entire message (including the header) as well as individual parts of the header.

The **List Command** expects you to enter one of the commands you see below. Those commands and their meanings are:

- t Lists the "To" field from the header.
- c Lists the "Cc" field from the header.
- b Lists the "Bcc" field from the header.
- h Lists the entire header.
- m Lists body of the message.
- a Lists both the full header and the message body.

To use the **List Command**, enter the letter **l** followed by one of the letters shown above. For example, to look at the "To" field on the message we are creating:

```
sm> l t (RET)
```

```
To: carole, white
```

```
sm>
```

## Looking at the Header, Continued

To look at the entire header follow the example below. Notice that Sndmsg changes around the entries you've filled in, and adds three new entries, namely the date and time the message was created, your name and address, and a message identification number (so that mail can be recovered if it is lost).

The first line of the header (X-ST-...) will appear if you *have* asked for a return receipt and will not appear if you have not.

```
sm> l h (RET)
```

```
X-ST-Return-Receipt-Requested:
```

```
Date: Mon, 29 Oct 84 09:30:51 PST
```

```
From: smith (rix jim) @ rix.dol
```

```
Message-Id: <841029093042.033@lbl-h.arpa>
```

```
Subject: We have received the manuals
```

```
To: carole, white
```

```
cc: virginia, postle @ rix, jones @ rx, nadel
```

```
Bcc: pasek @ etadc
```

```
sm>
```

---

## **Composing the Message**

---

Sndmsg gives you the tools you need to handle your outgoing correspondence tasks, whether you're dashing off a quick note or putting together a formal annual report.

You can take full advantage of other tools on the computer when composing your message. Report listings from programs such as Datatrieve, the results of statistical analysis, or the output from text formatters can be included in your message. Any editing changes you need to make to the message can be made from within Sndmsg.

At the same time, Sndmsg makes it easy for you to send off small messages without even using an editor.

Sndmsg's composition commands include:

- a     "Append" a message to the header.
  
- e     Create or edit a message with the editor of your choice.
  
- r     Read the contents of a file into your message.

## Composing the Message, Continued

### Small Messages - The Append Command

You can use any of the Sndmsg composition commands to create your message, but the Append Command is the one used most often to create small messages because the command lets you type in the text without using an editor.

However, as you will see on the next page, even small messages contain errors that need to be fixed, in which case you'll have to use an editor.

Thus far in the example, the message's header has been filled out and the user is now ready to compose the message.

Use the Append Command by entering the letter **a** when you see the `sm>` prompt. Type in the text of the message.

When you are through, enter a period (.) on a line by itself. Sndmsg will then return you to the `sm>` prompt from where you can send the message, unless you want to make changes.

```
sm> a (RET)
```

```
New copies of the MSG manuals are available in my office. (RET)  
(Room 3239) Please come by and get them. (RET)
```

```
• (RET)
```

```
sm>
```

## Composing the Message, Continued

### Editing Messages - The Edit Command

If you feel more comfortable working within the confines of an editor, you can create your entire message there. Alternatively, the editor can be used to make changes to text that has been entered with the **Append Command**, as shown below.

In this example, the message is edited with **EDT**, and the room number is changed from "3239" as it was incorrectly entered, to "3238." Use the **Edit Command** by entering the letter **e**. **Sndmsg** will automatically place your message in the editor's buffer and wait quietly until you are finished using the editor. After you have made the changes, exit from the editor in the normal manner.

At that point you can send the message or go back and make changes to the header, if you feel it's necessary. If you want to edit the message again, simply retype the **Edit Command**.

```
sm> e (RET)

1 New copies of the MSG manuals are available in my office.

*c

New copies of the MSG manuals are available in my office.
(Room 3238) Please come by and get them.

~Z
*exit (RET)

RIX$SYS:[TOOLS.TMP]T0015002F.SM4;1 2 lines

sm>
```

## Sending the Message - The Send Command

Once the message satisfies you, and any errors you want to fix have been corrected, you can send the message. This is done with the **Send Command** which is used by entering the letter **s** when you see the **sm >** prompt.

*Hint: If you've made substantial changes to the message header or body, you might want to list the entire message before sending it.*

```
sm > l a (RET)
```

```
X-ST-Return-Receipt-Requested:
```

```
Date: Mon, 29 Oct 84 09:30:51 PST
```

```
From: smith (rix jim) @ rix.doi
```

```
Message-Id: <841029093042.033@lhl-h.arpa>
```

```
Subject: We have received the manuals
```

```
To: carole, white
```

```
Cc: virginia, postle @ rix, jones @ rx, nadel
```

```
Bcc: pasek @ etadc
```

```
New copies of the MSG manuals are available in my office.  
(Room 3238) Please come by and get them.
```

```
sm >
```

Sndmsg will ask you to confirm that you want the message sent. This confirmation is requested from you to keep you from accidentally sending the message before you're ready.

After the message has been sent, Sndmsg will turn control back to MSG and you will be greeted with the **<-** prompt.

```
sm > s (RET)
```

```
Are you sure? [y/n] y (RET)
```

```
<-
```

---

## **Incorporating VMS Files into Your Message**

---

The ability to incorporate all manner of data, ranging from the results of a statistical package to a corporate report on management, greatly increases the mail system's usefulness to you and those you work with.

On the DCN, all data is kept in files. These files can comprise an entire MSG message, or be included as part of one.

As part of a message, you can annotate these files as much as you like, or leave off any comments altogether.

## Incorporating VMS Files into Your Message, Continued

In this example, the user sends the announcement of a meeting, and includes a copy of the agenda. Since the agenda was created with the Software Tools formatter, Format, the text itself resides in a VMS file named "agenda.txt."

As you can see from the example, the meeting announcement is typed in using the **Append Command**. The agenda is read in with the **Read Command** which is used by entering the letter **r**. The command requires that you enter the name of the file you want read into the message:

```
<- sndmsg [type SPACE to to confirm] (Space)
```

```
To: white,brown (RET)
```

```
To: (RET)
```

```
Cc: (RET)
```

```
Bcc: (RET)
```

```
Subject: Staff Review Meeting (RET)
```

```
Do you want a return receipt? [y/n] y (RET)
```

```
If you need help, type h
```

```
sm> a (RET)
```

```
We're going to have a staff review meeting on Wednesday at 3:00. (RET)
```

```
I'm enclosing a copy of the agenda. Please send me any comments you (RET)  
have. Thanks. (RET)
```

```
• (RET)
```

```
sm> r agenda.txt (RET)
```

```
sm>
```

## Incorporating VMS Files into Your Message, Continued

The **R**ead Command will not display the contents of the file. Therefore, it makes sense to list the entire message before you send it, checking to make sure that you've read in the correct file:

```
sm> l a (RET)
```

X-ST-Return-Receipt-Requested:

Date: Tue, 30 Oct 84 09:40:52 PST

From: smith (rix jim) @ rix.dol

Message-Id: <841030094042.003@rix.dol>

Subject: Staff review meeting

To: white,brown

We're going to have a staff review meeting on Wednesday at 3:00.  
I'm enclosing a copy of the agenda. Please send me any comments you have. Thanks.

### Staff Review Meeting Agenda

8:30	Introduction of Meeting
8:40	Description of Current Staff Duties (General)
10:00	Budget Evaluation
11:00	Presentation of Proposed Reorganization Methodology
12:00	Lunch
1:30	Discussion of Reorganization Impact
4:30	Adjournment

```
sm> s (RET)
```

```
Are you sure? [y/n] y (RET)
```



---

## Answering Your Mail

---

Some of your messages will require that you send back an answer. The **Answer Command** does just that, automatically addressing the message, and supplying a subject field.

When you issue the **Answer Command**, MSG will place you in Sndmsg. All of the Sndmsg commands shown earlier in this manual are at your disposal. For example, you can use the header commands described in the section entitled "Changing the Header" to send your answer to other people; you can incorporate the content of VMS files; or use the editor to make changes to the message's text.

To show you how to use this command, we will assume that Mr. Smith has received an answer to the message he sent in the preceding example. In that answer, Mr. White requests some changes to the meeting agenda. This example shows you the commands Mr. Smith uses to answer the message and, at the same time, send a copy to Mr. Brown.

```
$ msg (RET)
```

```
Software Tools MSG System
type ? for help
type # for news
type % for intro
```

```
1 messages / 1 new
```

```
1 658 30-Oct-1984 white Staff review Meeting
```

```
<- type (message sequence) 1 (RET)
```

```
(message 1, 289 characters)
```

```
Date: Tue, 30 Oct 84 09:56:15 PDT
From: white (rix sam) @ rix.dol
Message-Id: <840606095610.006@rix.dol>
Subject: Staff review Meeting
To: smith
```

```
I think we'll need to allot some time to draft up a new organization chart.
```

```
<-
```

## Answering Your Mail, Continued

To use the Answer Command, enter the letter **a**. MSG will ask you for a message number, which in this case is **1**. After you have specified a message number, MSG will automatically place you in Sndmsg:

```
<- a (RET)
<- answer message number: 1 (RET)
If you need help, type h

sm> a (RET)
White: (RET)
You're right. I've made some time at the end - this note is going to Brown too. (RET)
• (RET)

sm> r agenda.txt (RET)

sm> t (RET)
To: brown (RET)
To: (RET)

sm> l a (RET)

Date: Wed, 31 Oct 84 07:26:15 PST
From: smith (rlx jlm) @ rlx.dol
Message-Id: <841031072815.00c@rlx.dol>
Subject: Re: Staff review Meeting
In-Reply-To: Your message <841030121056.047@rlx.dol> dated 30-Oct-1984
To: <white>,brown

White:
You're right. I've made some time at the end - this note is going to Brown too.

                        Staff Review Meeting Agenda

8:30          Introduction of Meeting
8:40          Description of Current Staff Duties (General)
10:00         Budget Evaluation
11:00         Presentation of Proposed Reorganization Methodology

12:00        Lunch

1:30          Discussion of Reorganization Impact
4:10          Design of Draft Organization Chart
5:00          Adjournment

sm> s (RET)
Are you sure? [y/n] y (RET)
<-
```

---

## Forwarding Messages

---

If you receive a message that you feel someone else should see, you can use the **F**orward Command to send it to them.

The **F**orward Command automatically places you in **Sndmsg**, which means that the commands that change the header, list the message, or include attachments can be used.

The first thing **Sndmsg** will do is ask you for the names of the people to whom you want the message sent. You should enter the names just as you would when sending a message<sup>12</sup>. **MSG** will automatically supply the subject entry.

If you wish to annotate the message before sending it, you can use either of the two **Sndmsg** commands:

- a      "Appends" your annotation to the *end* of the message;
- i      "Inserts" your annotation to the *beginning* of the message;

In this example, Mr. Smith receives a message from Ms. Black about a change in the meeting schedule; he then forwards the message to Mr. White and Mr. Brown:

---

<sup>12</sup> See the section entitled "Filling Out the Header - the Envelope."

## Forwarding Messages, Continued

Use the **Forward Command** by entering the letter **f**. You will be asked to supply a message sequence and can use any of the four basic message sequences described in the section entitled "Reading Messages."

```
$ msg (RET)

                                Software Tools MSG System
                                type ? for help
                                type # for news
                                type % for intro

1 messages / 1 new
  1  230  31-Oct-1984 black  Schedule Change

<- type (message sequence) 1 (RET)

(message 1, 230 characters)
Date:   Wed, 31 Oct 84 09:56:15 PDT
From:   black (rix samantha) @ rix.dol
Message-Id: <840526095610.006@rix.dol>
Subject: Schedule Change
To:     smith

The conference room is not available on Wednesday; we'll need to reschedule
for Thursday.

<- forward (message sequence) 1 (RET)

To: smith,brown (RET)
cc: (RET)
Bcc: (RET)

Do you want a return receipt? [y/n] n (RET)

If you need help, type h

sm> i (RET)

Thought you might like to see this (RET)
• (RET)

<-
```

## Forwarding Messages, Continued

Use the **List** Command to look at the entire message before you forward it:

```
sm> l a (RET)
```

```
Date: Wed, 31 Oct 84 10:32:35 PST
From: smith (rix jim) @ rix.dol
Message-Id: <840526095610.006@rix.dol>
Subject: Re: Schedule Change
To: white,brown
```

Thought you might like to see this

----- Start of forwarded message[s] -----

```
Return-path: <black@rix.dol>
Received: from rix.dol by lbl-h.arpa with DECNET ;
        Wed, 31 Oct 84 09:56:02 PST
Date: Wed, 31 Oct 84 09:50:57 PST
From: black (rix samantha) @ rix.dol
Message-Id: <841030104557.041@rix.dol>
Subject: Schedule Change
To: smith
```

The conference room is not available on Wednesday; we'll need to reschedule for Thursday.

----- End of forwarded message[s] -----

```
sm> s (RET)
```

```
Are you sure? [y/n] y (RET)
```

```
<-
```



---

## Moving Your Mail to a File — Getting a Paper Copy

---

There are several reasons for wanting to move a message out of the mail system and onto paper or a VMS file. You may wish to keep a paper file of all the messages you receive about a certain topic, or maybe move your entire mail box or folder to paper so that you can read it at a more convenient time or place.

Alternatively, some of your messages may contain program listings or procedures. You will need to move these from the mail box into a VMS file to use them.

There are two commands that move messages out of your mail box:

The **List Command** is used to print a message on the line printer or a VMS file. The command requests that you supply it with a message sequence, which can be any of the four described earlier. Unlike the **Move Command**, **List** does *not* mark the messages for deletion. The result of the **List Command** gives you two things: First, **List** will print a cover page that contains the headers of each of the messages you specify. Following the cover page, each message will appear on a separate page. For example, suppose you had seven messages in your incoming mail box (or in a folder) and “listed” them to the line printer. You would receive a single page containing the message headers of all seven messages. The rest of the pages would contain the seven messages, each starting on a new page.

The **Put Command** is used to move messages to a VMS file. The messages will be placed on the file without any special formatting characters. If you are moving program listings or procedures, you will need to edit the VMS file to remove the message headers before you can use them. If you want to get a hard copy of a message you have moved with the **Put Command**, you will need to use the “print” command outside of MSG, at the DCL prompt, \$.

## Moving Your Mail to a File — Getting a Paper Copy, Continued

To use the List Command enter the letter l. MSG will ask you for a message sequence; you can use any of the four described earlier. You will then be asked for a file name. At this point you can specify either the line printer, or a VMS file.

In this example, the user lists all the messages in his incoming mail box on the line printer. Notice that the acronym for the line printer is the characters, lp: (be sure to include the colon).

```
$ msg (RET)

                                Software Tools MSG System
                                type ? for help
                                type # for news
                                type % for intro

2 messages / 2 new
  1  427  31-Oct-1984 white  Schedule Change
  2  304  31-Oct-1984 black  Conference Room

<- list (message sequence) all

into file name: lp: (RET)

<-
```

The output the user will receive when he goes to the line printer will look like this:

## Moving Your Mail to a File — Getting a Paper Copy, Continued

1 427 31-Oct-1984 white Schedule Change  
2 304 31-Oct-1984 black Conference Room

(message 1, 289 characters)  
Date: 427, 31 Oct 84 12:56:15 PDT  
From: white (rix sam) @ rix.dol  
Message-Id: <840606095610.006@rix.dol>  
Subject: Schedule Change  
To: smith

Can you send me the exact time, room number, and date of the meeting when the schedule finally settles down?

Thanks,

Sam

(message 2, 304 characters)  
Date: 427, 31 Oct 84 12:59:18 PDT  
From: black (rix samantha) @ rix.dol  
Message-Id: <850607094609.006@rix.dol>  
Subject: Conference Room  
To: smith

Jim,

The conference room has been reserved for you from 8:00am until 6:00 pm on Thursday.

You will be using the Oak Room — it has kitchen facilities.

Samantha

## Moving Your Mail to a File — Getting a Paper Copy, Continued

The **Put** Command is used to move messages to a VMS file. Use it by entering the letter **p**. The command will ask you to provide a message sequence and a VMS file name. You can use any of the four basic message sequences, and, if you are not familiar with VMS file names, you should refer to the "Introduction to DCN" booklet.

```
$ msg (RET)
```

```
Software Tools MSG System
type ? for help
type # for news
type % for intro
```

```
2 messages / 2 new
```

```
1 427 31-Oct-1984 white Schedule Change
2 304 31-Oct-1984 black Conference Room
```

```
<- put (message sequence) 2 (RET)
```

```
into file name: meeting.msg (RET)
```

```
<-
```

To print this message on the line printer, the user must return to DCL level, and use the **print** command. He must specify the name of the file that he wants printed, i.e., the file that contains the message. DCL's response tells the user that the file has been dispatched to the line printer:

```
$ print meeting.msg (RET)
Job 4008 entered on queue sys$print
$
```

---

## Return Receipts

---

Return receipts are used whenever you want confirmation that the people you have sent the message have, in fact, read the message.

You request the return receipt at the time you create the message. As each recipient reads the message, the MSG system automatically sends a message back to you. Included in the message is the date and time the message was read.

A return receipt will be returned to you from *all* people on the message's address list, including those on the "To," "Cc," and "Bcc" lists.

## Requesting a Return Receipt

As you fill out the message header, MSG will ask if you want a return receipt. If you answer "yes," a receipt will be sent to you by each person reading the message.

The return receipt specification can be reset by using the **Define Return Receipt Command** as described in the section entitled "Changing the Header."

```
<- sndmsg [type SPACE to to confirm] (Space)
```

```
To: agazzi (RET)
```

```
To: (RET)
```

```
Cc: virginia.postle@rix,jones@rx (RET)
```

```
Cc: (RET)
```

```
Bcc: pasek@etadc (RET)
```

```
Bcc: (RET)
```

```
Subject: manuals (RET)
```

```
Do you want a return receipt? [y/n] y (RET)
```

```
If you need help, type h
```

```
sm >
```

## Sending Back a Return Receipt

MSG uses the message status **R** to mark those messages that have a return receipt attached to them, as can be seen in the first column of the example shown below.

When you read such a message, a receipt is sent automatically. After the message has been displayed, MSG will print a message on your terminal telling you that it is sending the return receipt. After a short pause, you can continue reading your mail:

```
$ msg (RET)

                                Software Tools MSG System
                                type ? for help
                                type # for news
                                type % for intro

1 messages / 1 new
NR  1  304  6-Jun-1984 green  report status

<- type (message sequence) 1 (RET)

(message 1, 304 characters)
Date:   Wed, 6 Jun 84 10:30:45 PDT
From:   green (lblh csam ethel) @ lbl-h.arpa
Message-Id: <846060010376.009@lbl-h.arpa>
Subject: report status
To:     smith

It is critical that we get a report status today.

Sending receipt - please wait for prompt

Submitting mail to MAILER

<-
```

## Reading the Return Receipt

Return receipts are placed in your incoming mail box as they are received. In the example below, you can see that the message header tells you that a return receipt has been sent by "THE-MAIL-SYSTEM." The body of the receipt includes the name of the person who has read your message in the "Sender" field in the message header. The body of the message includes the date and time the message was read:

\$ msg (RET)

Software Tools MSG System  
type ? for help  
type # for news  
type % for intro

1 messages / 1 new

N 1 488 6-Jun-1984 THE-MAIL-SYSTEM Automatic receipt

<- type (message sequence) 1 (RET)

(message 1, 488 characters)

Date: Wed, 6 Jun 84 10:55:39 PDT  
From: THE-MAIL-SYSTEM@lbl-h.arpa  
Sender: simth (lblh csam jim)  
Message-Id: <840828105539.007@lbl-h.arpa>  
Subject: Automatic receipt  
To: <green>

This is an automatically generated acknowledgement of the receipt of your message. Excerpts from the message header follow:

Date: Wed, 6 Jun 84 10:45:58 PDT  
Subject: report status  
Message-Id: <840727091057.00b@lbl-h.arpa>

<-

---

## Creating and Using Mailing Lists, Malias

---

Mailing lists are used to assign a single name to a list of people's names and addresses. These single names, which are referred to as aliases, can be used in place of a series of individual names and addresses. It is easy to see that using the aliases can significantly reduce the amount of typing needed to address messages.

The list of aliases and the names and addresses they represent are kept in a file in your main directory called "**malias.**" which is created in much the same way as the "folders." file is created.

The **malias.** file must include:

- (1) an alias followed by a colon,
- (2) a list of the names the alias represents, and
- (3) a semicolon at the end of the list of names.

The example below shows you how to create the "malias." file by using the DCL Command, **create**. The **create** command is described in more detail in the section entitled "Creating File Folders."

```
$ create malias. (RET)

staff:brown,black,white; (RET)
software:schroeder,nadel,mccarthy; (RET)
review:brown,white; (RET)

^Z

$
```

## Using Mailing Lists in Sndmsg

The aliases created in the previous example can be used in place of names in the address fields "To," "Cc," and "Bcc."

In this example, a message is sent to the people listed on the alias "staff".

```
<- sndmsg [type SPACE to to confirm] (Space)
```

```
To: staff (RET)
```

```
To: (RET)
```

```
Cc: (RET)
```

```
Bcc: (RET)
```

```
Subject: Staff Review Meeting (RET)
```

```
Do you want a return receipt? [y/n] n (RET)
```

```
If you need help, type h
```

```
sm> a (RET)
```

```
Copies of the minutes from the staff review meeting are now available. (RET)
```

```
• (RET)
```

```
sm>
```

## Using Mailing Lists in Sndmsg, Continued

Sndmsg will automatically expand the alias to its full list of names before sending the message.

To verify that the message is being sent to the proper people, use the **List Command** to look at the entire message by specifying the **all** subcommand.

If you want to add names to the list of people to whom the message will be sent, use the commands described in the section entitled "Changing the Header."

```
sm> l a (RET)
```

```
Date: Thu, 7 Jun 84 09:40:52 PST
From: smith (rix jim) @ rix.dol
Message-Id: <841030094042.003@rix.dol>
Subject: Staff review meeting
To: brown,black,white
```

Copies of the minutes from the staff review meeting are now available.

```
sm> s (RET)
```

```
Are you sure? [y/n] y (RET)
```

## Examining Mailing Lists

The Sndmsg command **View** can be used to look at the definitions of your aliases.

This command must be used at the Sndmsg prompt, `sm >`

You can look at the definition of a specific alias, or ask to see the definitions of all your aliases.

To look at the definitions of all aliases, use the **View Command** followed by a space and an asterisk (\*):

```
sm > v *  
  
staff: brown,black,white;  
  
software: schroeder,nadel,mccarthy;  
  
review: brown,white;  
  
sm >
```

To look at the definition of a single definition, use the **View Command** followed by the alias:

```
sm > v staff  
  
staff: brown,black,white;  
  
sm >
```

---

## MSG Command Summary

---

**STARTING THE MAIL SYSTEM** — to start at the \$ prompt, type:

<code>\$msg</code>	to read the mail and access all electronic mail facilities
<code>\$sndmsg</code>	to send messages only

**LEAVE MSG** — in MSG, when you have the <- prompt

<i>Command Letter</i>	<i>Explanation</i>
<code>e[xit]</code>	the recommended way of leaving the MSG system; exits the message system and UPDATES message file
<code>q[uit]</code>	leaves the message system WITHOUT updating message file

**HELP** — in MSG and SNDMSG

<i>Command Letter</i>	<i>Explanation</i>
<code>?</code>	in MSG provides list of all MSG commands, in SNDMSG provides list of all sndmsg commands
<code>h[elp]</code>	in SNDMSG provides help on SNDMSG commands — invoked at the sm> prompt (synonym for ?)
<code>i[nformation]</code>	at the <- prompt provides an explanation of specified MSG commands

**READ MESSAGES** — in MSG when you have the <- prompt

<i>Command Letter</i>	<i>Explanation</i>
<code>g[o]</code>	goes to a specified message and prints it on the terminal
<code>t[ype]</code>	types message or a number of specified messages on the terminal
<code>t[ype]a[ll]</code>	types all messages on the terminal
<code>t[ype]s[ubject]</code>	types all messages by specified subject matter
<code>t[ype]f[rom]</code>	types all messages from specified sender
<code>b[ack]</code>	backs up to last message and prints it on the terminal
<code>n[ext]</code>	goes to the next message specified and prints it on the terminal
<code>c[urrent]</code>	displays the number of the current message, the total number of messages, and the current filename
<code>h[earer]</code>	prints the headers of a message
<code>h[earer]a[ll]</code>	prints the headers of all messages
<code>h[earer]s[ubject]</code>	prints the headers of all messages by specified subject matter
<code>h[earer]f[rom]</code>	prints the headers of all messages from specified sender
<code>s[end]</code>	sends a message to the MSG Mailer

---

## MSG Command Summary, Continued

---

**COMPOSE MESSAGES** — in SNDMSG, when you have the sm> prompt

<i>Command Letter</i>	<i>Explanation</i>
(Space)	hit space bar
e[dit] edit	invokes the EDIT word processing system
a[ppend]	appends text to the message
r[ead] filename	reads into the message, material from another file Can be used in the send, answer and forward message modes
j[ump]	permits the user to jump into the Software Tools Shell (Move back to MSG from the Shell by Control Z)
l[ist]t[o]	prints the current TO field
l[ist]c[opy to]	prints the current carbon copy (cc) field
l[ist]b[lin d cc to]	prints the current blind carbon copy (bcc) field
l[ist]h[ead er]	prints the contents of the entire header
l[ist]m[ess age]	prints the text of the message
l[ist]a[ll]	prints both headers and message contents

**SEND MESSAGES** — in SNDMSG, when you have the sm> prompt

<i>Command Letter</i>	<i>Explanation</i>
(Space)	means hit space bar
s[end]	sends message to MSG Mailer
i[nsert]	inserts text at top of forwarded messages
a[ppend]	appends text at bottom of forwarded messages
t[o]	adds names to the TO field of the header
b[lin d cc]	adds names to blind carbon copy (bcc) field in header
c[opy]	adds names to carbon copy (cc) field in header
d[efine] r y	changes return receipt request to yes
d[efine] r n	changes return receipt request to no
d[efine] s new subject	changes the subject field of the message to specified "new subject"
v[iew] alias name	prints the names in the specified alias file on the terminal
v[iew]*	prints the names and contents of all alias files on the terminal
m[alias]	A MSG alias (malias) permits the user to use one name instead of a number of names when sending the same message to a number of persons.

---

## MSG Command Summary, Continued

---

**ANSWER, FORWARD AND SEND MESSAGES** — in MSG, when you have the <- prompt

<i>Command Letter</i>	<i>Explanation</i>
<b>a</b> [nswer]	answers specified message
<b>f</b> [orward]	forwards specified message
<b>s</b> [end]	sends specified message

**PRINT, MOVE AND FILE MESSAGES** — in MSG, when you have the <- prompt

<i>Command Letter</i>	<i>Explanation</i>
<b>l</b> [ist]	lists message(s) on a file so that they can be printed
<b>l</b> [ist] <b>number lp:</b>	lists message(s) on the line printer
<b>m</b> [ove]	moves message(s) into another file and deletes message from current file
<b>p</b> [ut]	puts message(s) into another file

**DELETE MESSAGES** — in MSG, when you have the <- prompt

<i>Command Letter</i>	<i>Explanation</i>
<b>d</b> [elete]	deletes message from file by number(s)
<b>d</b> [elete] <b>a</b> [ll]	deletes all messages from file
<b>d</b> [elete] <b>s</b> [ubject]	deletes messages by specified subject
<b>d</b> [elete] <b>f</b> [rom]	deletes messages from specified senders
<b>o</b> [verwrite]	overwrites the message file by deleting specified messages and renumbers the messages left in the file

**RECOVER DELETED MESSAGES** — can only be done before exiting from MSG

<i>Command Letter</i>	<i>Explanation</i>
<b>u</b> [ndelete]	undoes the action of the delete command; undeletes messages from file by specific number(s)
<b>u</b> [ndelete] <b>a</b> [ll]	undeletes all messages
<b>u</b> [ndelete] <b>s</b> [ubject]	undeletes messages by specified subject
<b>u</b> [ndelete] <b>f</b> [rom]	undeletes messages from specified sender

---

## MSG Command Summary, Continued

---

**MESSAGE SEQUENCES** — in MSG, when you are asked for a message sequence

<i>Command Letter</i>	<i>Explanation</i>
<b>a</b> [ll]	Refers to <i>all</i> messages
<b>c</b> [urrent]	Refers to the <i>current</i> message
<b>d</b> [eleted]	Refers to all <i>deleted</i> messages
<b>f</b> [rom]	Refers to all messages <i>from</i> a specific sender
<b>n</b> [ew]	Refers to all <i>new</i> messages
<b>s</b> [ubject]	Refers to text in the <i>subject</i> field of all messages
<b>t</b> [ext]	Refers to words in the <i>text</i> body of all messages
<b>u</b> [ndeleted]	Refers to all <i>undeleted</i> messages
<b>RET</b>	Refers to the <i>current</i> message

<i>Numeric Sequences</i>	<i>Explanation</i>
Single Number	Refers to a single message
Series of Numbers	Refers to a series of messages, separated by commas
Range of Numbers	Refers to a range of messages, separated by a dash
Combination	Refers to a combination of a single message number, a series of message numbers, and a range of message numbers

---

## Index

---

- RET** 4
- <-**, 6
- sm>**, 51
- (Space)**, 4
- <Blank Line>**, 4
- ?**, 8
- %**, 8
- #**, 9
- a[append]**, 58,67
- a[answer]**, 65
- b[lindcc]**, 48
- c[copy]**, 48, 53
- d[efine]**, 52,54
- d[delete]**, 29—30
- e[dit]**, 59
- e[xit]**, 12
- f[orward]**, 67
- h[eader]**, 25
- h[elp]** (in Sndmsg), 50
- i[insert]**, 67
- i[nformation]**, 9
- l[ist]**, 55
- m[ove]**, 38
- o[verwrite]**, 14
- q[uit]**, 13
- r[ead]** (in Sndmsg), 62
- t[o]**, 48,53
- t[ype]**, 18
- u[ndelete]**, 33
- v[iew]all**, 45
- v[iew]+folder-name**, 45
- v[iew] alias name**, 82
- v[iew]\***, 82
- Accessing Messages** —
  - Stored in Folders, 41
- Adding Names to a** —
  - Message Address, 52
- Addresses**, 48
- Addressing Messages**, 48
- Aliases**,
  - Creating, 79
  - Looking at, 82
- All, the Message Sequence**, 15,24
- Answering Your Mail**, 65

## Index, Continued

### **Attaching a File —**

to Your Message, 61

### **Changing the Header, —**

When Creating, 52

### **Composing a Message, 57**

### **Create, —**

the DCL Command, 37

### **Creating File Folders, 36**

### **Creating and Using —**

Mailing Lists, Mallas, 79

### **DCL, 5,79,37,79**

### **Deleting Messages,**

All Messages, 31

By Number, 30

By Sender, 30

By Subject, 31

### **Discarding Messages,**

See Deleting Messages

### **Editor,**

When Creating Messages, 59

### **Entering MSG Commands, 7**

### **Exit Command, 12**

### **File Folders —**

Creating, 36

### **Filing System for Messages, 35**

### **Filling out the Header —**

the Envelope, 51

### **Finding a Message,**

See Headers

### **Folders, 35**

### **Forwarding Messages, 67**

### **From Message Sequence, 21**

### **Getting a Paper Copy —**

of a Message, 71,72

### **Headers,**

for all Messages, 25

by Number, 26

by Sender Name, 27

by Subject, 27

the MSG Envelope, 16,48

### **Help,**

In Msg, 8

In Sndmsg, 50

## Index, Continued

### **Identifying a Message,**

See Message Sequences

### **Incorporating VMS Files into**

Your Message, 61

### **Leaving**

MSG, 11

Sndmsg, 51

### **Looking at the Header**

in Msg, See Headers

in Sndmsg, 55

### **MSG Command, 5**

### **MSG Command Summary, 83**

### **Mail Box, —**

What's In It, see Headers

### **Mailing Lists, 79**

### **Aliases, 79**

### **Message Headers,**

see Headers

### **Message Sequences,**

see Headers

### **Moving Messages —**

Into Folders, 38

Into a File, 38

### **Notations,**

Used in this Manual, 4

### **Notification of Delivery,**

see Return Receipts

### **Numeric Message Sequences, 18**

### **Overwrite Command, 14**

### **Print,**

DCL, 74

### **Putting Messages into Folders,**

See Moving Messages

### **Putting Messages into a File, 74**

### **Quit Command,**

In MSG, 13

### **Quit Command,**

In Sndmsg, 51

### **Reading**

All Messages, 24

Messages Stored in Folders, 41

Messages by Number, 18

Messages by Sender Name, 21

Messages by Subject, 23

## Index, Continued

**Reading a File into a Message, 61**

**Recovering Deleted Messages, 33**

**Resetting, 52**

Message Header, 52

Message Subject, 54

Return Receipt, 52

**Responding to a Message,**

See Answering Your Mail

**Return Receipts,**

Reading, 78

Requesting, 76

Sending, 77

with Exit, 12—14

with Qult, 12—14

**Rewriting Your Mail Box or**

Folder, see Overwrite

**Saving Messages,**

see Folders

**Sending Messages, 47**

**Sndmsg, 47**

**Specifying a Message**

by Number, 18

by Sender Name, 21

by Subject, 23

**Starting MSG, 5**

**Stopping MSG, 12—14**

**Subject Message Sequence, 23**

**Throwing Messages Away, 29**

**Typing Messages,**

See Reading Messages

**Undeleting Messages, 33**

**Viewing and Adding Folders, 45**

**Viewing and Creating Aliases, 79**

**Writing a Message,**

See Sending Messages

This report was done with support from the Department of Energy. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the Department of Energy.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

TECHNICAL INFORMATION DEPARTMENT  
LAWRENCE BERKELEY LABORATORY  
UNIVERSITY OF CALIFORNIA  
BERKELEY, CALIFORNIA 94720