SIGMA Message Service Reference Manual ISI/TM-78-11.2 Version 2.3

AD A072840

12 B.S.

ISI/TM-78-11.2
June 1979
ARPA ORDER NO. 2223

LEVEL III

DARPA NAVY CINCPAC

Military Message Experiment

SIGMA Message Service Reference Manual

Version 2.3

DDC RECEIVED AUG 15 1979 C

Jeff Rothenberg

DDC FILE COPY

This document has been approved for public release and sale; its distribution is unlimited.

INFORMATION SCIENCES INSTITUTE
UNIVERSITY OF SOUTHERN CALIFORNIA
4676 Admiralty Way/Marina del Rey/California 90291
(213) 822-1511

79 08 1 - 012

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

(14) 1. REPORT NUMBER: ISI/TM-78-11.2 ✓ | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER

(6) 4. TITLE (and Subtitle): SIGMA Message Service: Reference Manual, AD68178 Version 2.3.

(9) 5. TYPE OF REPORT & PERIOD COVERED: Technical Manual

6. PERFORMING ORG. REPORT NUMBER

(10) 7. AUTHOR(s): Jeff Rothenberg

(15) 8. CONTRACT OR GRANT NUMBER(s): DAHC 15-72-C-0308 ARPA Order-2223

9. PERFORMING ORGANIZATION NAME AND ADDRESS: USC/Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90291 ✓

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS: Defense Advanced Research Projects Agency, 1400 Wilson Blvd., Arlington, VA 22209

(11) 12. REPORT DATE: June 1979

13. NUMBER OF PAGES: 79

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): —

(12) 84p.

15. SECURITY CLASS. (of this report): Unclassified

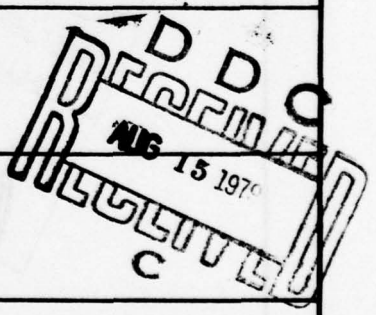
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This report approved for public release and sale; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

—



18. SUPPLEMENTARY NOTES

—

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

AUTODIN, editing, interactive terminal, man-machine interaction, message processing, MME (military message experiment), user interface

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

(OVER)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

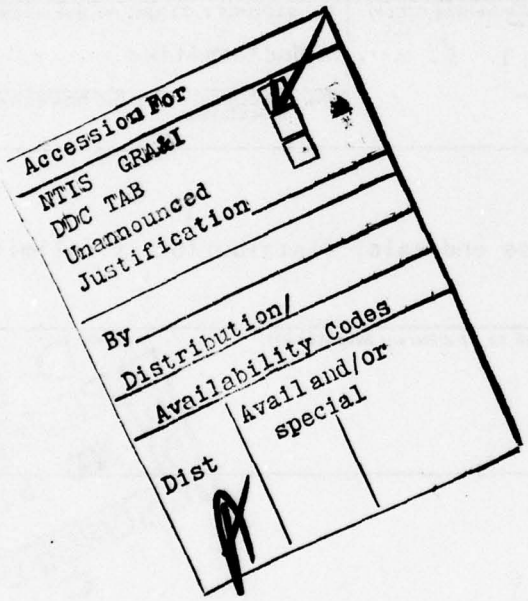
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

407 952

LB

20. ABSTRACT

The SIGMA Message Service is an automated message processing service provided for a test of automated military message processing. Through the service, procedures that are presently both manual and semi-automated are completely automated; these include everything from message composition through message receipt. The SIGMA Message Service allows work on four kinds of objects: messages, files of messages, text items, and selectors. Each of the four types is explained, and instructions applicable to each are described in detail. This manual also describes, from the point of view of a novice user, the effect of each instruction that the user may issue to the service.



ISI/TM-78-11.2
June 1979
ARPA ORDER NO. 2223

DARPA NAVY CINCPAC
Military Message Experiment

SIGMA Message Service Reference Manual

Version 2.3

Jeff Rothenberg

INFORMATION SCIENCES INSTITUTE

UNIVERSITY OF SOUTHERN CALIFORNIA

4676 Admiralty Way/ Marina del Rey/ California 90291
(213) 822-1511

THIS RESEARCH IS SUPPORTED BY THE ADVANCED RESEARCH PROJECTS AGENCY UNDER CONTRACT NO. DAHC15 72 C 0308, ARPA ORDER NO. 2223.

VIEWS AND CONCLUSIONS CONTAINED IN THIS STUDY ARE THE AUTHOR'S AND SHOULD NOT BE INTERPRETED AS REPRESENTING THE OFFICIAL OPINION OR POLICY OF ARPA, THE U.S. GOVERNMENT OR ANY OTHER PERSON OR AGENCY CONNECTED WITH THEM.

THIS DOCUMENT APPROVED FOR PUBLIC RELEASE AND SALE: DISTRIBUTION IS UNLIMITED.

TABLE of CONTENTS

I.	Outline of Service Facilities	1
	General Service Descriptions	1
	Instructions for Files	1
	Instructions for Messages	2
	Instructions for Text	2
	Instructions for Selectors	3
	Miscellaneous Instructions	3
II.	Introduction	4
	Overview of SIGMA Service	4
	SIGMA Service Documentation	5
	Service Features	6
III.	Message Formats	8
	AUTODIN Messages	8
	Internal Messages	8
IV.	File System	10
	Special Files	11
	Action Log	12
	Date Files	13
V.	Alerts	14
VI.	Message Drafting	15
	Initiating the Message Draft	15
	Other Features	17
	Memoranda	17
	Informal Notes	18
	References	19
	User Lists	19
	Coordination	19
	Transmission	19
	Correcting Errors	19
VII.	Message Review and Coordination	20
	The Chop List	20
	Message Review	21
	Multiple Coordination Cycles	22
VIII.	The Terminal Screen	23
IX.	Terminal Keys	26
X.	Function Keys	28
XI.	Parameter Descriptions	29
XII.	Instruction Descriptions	35

I. OUTLINE OF SERVICE FACILITIES

GENERAL SERVICE DESCRIPTIONS

The SIGMA Message Service is documented in several forms. The printed documentation and the on-line HELP documentation are essentially identical. They consist of the general description chapters and the instruction descriptions outlined below. In addition, lessons are available on-line (see the Miscellaneous instruction descriptions below) which will allow you to learn a number of different aspects of the SIGMA message service, and do exercises associated with individual lessons.

The material below is organized by service functions. There are four Objects (kinds of things) with which you will be working using the SIGMA message service. These objects are Messages, Files of messages, Text items and Selectors. A Text item is any arbitrary group of words composed by the user, such as an AIG list. A Selector is a single field, or a number of message fields defined as the basis for the retrieval of messages when reading a file. Instructions which allow you to perform a number of useful operations on these four objects are described in detail in the Instruction Descriptions section. In the outline below, however, we simply name the instructions needed to perform a range of tasks for each object.

To use the outline below, find the object on which you are working, then under that group, *find the instruction needed* to perform the desired operation. For further elaboration of a particular instruction, find its description in the Instruction Descriptions section, which details the forms of the instructions and gives examples. In addition to the instructions on the four objects of the service (Messages, Files, Text and Selectors) there are miscellaneous instructions for logging on and off, and changing your identification.

Note that some of the following instructions must be typed in, while others are entered by a single Function key stroke.

FILES

ABORT -- Terminate without updating.
AUGMENT -- Add to current file display by selector.
BACKUP -- Re-display file to previous selection criteria.
CLEAR -- Erase the view window.
COMMENT -- Comment on an entry in the displayed file.
CREATE -- Create a new file.
DELETE -- Delete an entry from the displayed file, or delete a named file entirely.
DISPLAY -- Display the entries in a file.
EMPTY -- Delete all the entries in the specified file.
FILE -- Place the identified message into the named file.
FILE-DIRECTORY -- Show all this user's files.
FIND -- Find the specified entry and scroll the entries to the specified one.
FIND STRING -- Find the specified string of characters in an entry and scroll to that entry.
FINISH -- Terminate the current file and update.

GET -- Access another's file.
HIGHLIGHT -- Highlight Subject line of indicated entry.
KEYWORD -- Assign keywords to file entries as basis for retrieval.
MOVE -- Move the entry out of the current file into another file.
PRINT -- Print the File Directory or the open file.
RESTORE -- Restore a deleted entry from the displayed file or restore a previously deleted file.
RESTRICT -- From currently selected entries, display only those with selection criteria.
ROUTE -- Specify routing for a number of entries at once (ROUTE compounds ACTION, FORWARD, FILE and DELETE ENTRY LIST into a single operation).
SHOW-FILE -- Show the current open file in the Display window.
SORT -- Order the entries by DTG for the specified file.
VIEW -- Put entry or File Directory in view window. Cannot be edited.

MESSAGES

ABORT -- Terminate without updating.
ACTION -- Assign action to another user and make an entry in the Action Log.
CHOP YES or NO -- Sign off on a message draft.
CLEAR -- Erase the view window.
COMMENT -- Comment on a field of a displayed message.
COORDINATE -- Send a message out for chop to the people on the chop list.
COPY -- Create a new copy of a message for editing.
CREATE -- Create a new message and display the blank message form for editing.
DISPLAY -- Display the identified message for editing.
FILE -- File a copy of the identified message into a file.
FINISH -- Terminate the current message.
FORWARD -- Send a copy of a message to another user.
HIGHLIGHT -- Highlight indicated text of a Transmitted message.
PICKUP -- Place the marked text in a text object and delete it from display.
PRINT -- Print message on the printer.
PUT -- Put the text object at the marked place in the display.
READDRESS -- Create a special Readressal request message and display it for editing.
RELEASE -- Release the current message for transmission.
REPLY -- Create and display a reply to the current message.
SHOW-MESSAGE -- Show the current open message in the Display window.
UPDATE -- Perform name recognition and text formatting.
VIEW -- Put message in view window. Cannot be edited.

TEXT

ABORT -- Terminate without updating.
CLEAR -- Erase the view window.
COPY -- Create a new copy of text for editing.
CREATE -- Create a new text object.
DELETE -- Delete the named text object.
DISPLAY -- Display the named text object for editing.
FIND STRING -- Place the cursor at the first occurrence of the specified string.
FINISH -- Terminate the current text object and update editing.
GET -- Access another's text.

PICKUP -- Place the marked text in a text object and delete it from the display.
PRINT -- Print Text Directory or the text object on the printer.
PUT -- Put the text object at the marked place in the display.
RECLASSIFY -- Change the classification of a text object.
RESTORE -- Restore a previously deleted text object (opposite of Delete).
SAVE -- Save editing up to this point.
SHOW-TEXT -- Show the current open text object in the Display window.
TEXT-DIRECTORY -- Show all this user's text objects.
UPDATE -- Perform text formatting.
VIEW -- Put text in view window. Cannot be edited.

SELECTORS

AUGMENT -- Add to current display by selector.
BACKUP -- Redisplay file to previous selection criteria.
CREATE -- Create a new selector with the specified criteria, or save the selector for the current file entries, if no criteria are specified.
DELETE -- Delete a named selector.
GET -- Access another's selector.
PRINT -- Print the named Selector or the Selector Directory.
RESTORE -- Restore a previously deleted selector (opposite of Delete).
RESTRICT -- From currently selected entries, display only those with selection criteria.
SELECTOR-DIRECTORY -- Show all this user's selectors.
VIEW -- Put selector in view window. Cannot be edited.

MISCELLANEOUS

ALERT ON/OFF -- Shows the Alert List in place of the view window.
CLEAR -- Erase the view window.
CURRENT-ENTRY -- Make the marked entry in a file the current entry.
EXERCISE -- Start an exercise from within a lesson.
GO-TO-NEXT -- Make the next entry in a file the current entry.
HELP -- Shows the user an on-line version of this Reference Manual.
IDENTIFY -- Specify user name performing as office code.
LESSON -- Start on a lesson.
LOG OFF -- Log off of the SIGMA message service.
PRINT -- Print the displayed object or the viewed object.
PROMPT -- Shows the instructions and parameters SIGMA expects.
QUIT -- Exit from a lesson.
RESET ALERTS -- Makes Alert count 0 and clears Alert List.
SYSTEM NEWS -- Redisplays the News that is shown after Logging On.
SYSTEM STATUS -- Shows the status of other users of SIGMA.
VIEW -- Place named directory or displayed object into view window.

II. INTRODUCTION

OVERVIEW OF SIGMA SERVICE

The SIGMA message service is an automated message processing service provided for a test of automated military message processing. Through the service, procedures that are presently both manual and semi-automated are completely automated from message composition through message receipt. Users may prepare draft messages on-line by using standard message formats provided by the service. Text processing features offered by the service aid the user in composing and editing. The message review or chopping procedures are automated so that draft messages may be electronically dispersed to all reviewers simultaneously. Reviewers may approve, disapprove, comment and edit entirely through the automated service. The drafts, of course, are also returned to the drafter and/or sent to the releaser by the same electronic means. This should reduce the elapsed time needed for review during the Chop process.

Although the message handling procedures have been automated, users may at any time obtain printed copy of any information they possess within the service. Commonly, one might get a printed copy of messages referenced in a message to which one intends to reply.

Upon release, outgoing messages are delivered electronically to the AUTODIN system for further dispersal. Comeback copies are automatically sent to the internal distribution list. Likewise, incoming AUTODIN messages are received on-line automatically, as are messages forwarded for information and action. In addition to these formal AUTODIN messages, the service provides two other categories of messages. One is a formal Memo for official use within the headquarters, and the second is an Informal Note, for in-house unofficial use.

A difficult and time-consuming part of present procedures centers around storage, organization, and retrieval of messages. The service provides on-line files, conceptually similar to standard filing cabinets. Users may construct any number of on-line files and organize them in any way they desire. These files may contain arbitrary numbers of entries.

In addition to these personal files, users are automatically assigned other special files. Associated with each office code are special Pending files containing entries for messages and drafts awaiting attention. Entries are made there automatically when a message is sent to that office. Similar to these files, each user has a file for his own personal use labeled Mypending. (See the section on Special Files in IV. FILE SYSTEM.)

Files may be defined to be used as readboards such that they may be accessed by the necessary office codes. Lastly, the Action Log is an electronic file named ACTION_LOG. (See the section of Special Files in IV. FILE SYSTEM.)

When the user asks SIGMA to display a message which is old, he will often be told it has been Archived, and he is asked to Acknowledge Retrieval with the !!YES!! key. The message will be retrieved from tape, and a Retrieved entry will appear in the user's Pending file when this has been done.

In addition to messages and files of message entries, the service provides for the storage of arbitrary text. For example, users may construct personal AIG lists as text and insert them into the chop list of a draft message, in lieu of typing these office codes into each draft.

Messages, files, and pieces of text are called "objects." Objects are "opened" when they are worked with by the user, and "closed" either automatically when another object of the same type is opened, or when a FINISH instruction is used on it. The user may "open" or work on one of each type concurrently. Thus you may have a file open, a message open which you are displaying, and also a text object which you have been working on. You may not, however, have two files open simultaneously, two messages open, or two text objects open. For example, if you are displaying one message and attempt to display another, the system automatically closes the first message for you and opens and displays the second.

SIGMA SERVICE DOCUMENTATION

The SIGMA Message Service is an interactive message processing service provided for a test of automated military message processing. It is documented at the user's level in several forms. Here, we describe the printed Reference Manual and the on-line documentation, which are equivalent.

The Reference Manual and the on-line documentation mainly describe the effect of each instruction that the user may issue to the service. To show and explain the instruction forms, we use angle brackets and parentheses. Look at the instruction below, which causes the service to bring up a blank message form for the user to fill in. A more detailed explanation is presented later.

CREATE MESSAGE <MESSAGE TYPE> <SECURITY>

The words CREATE and MESSAGE are typed by the user. The words appearing in the angle brackets are not typed in exactly, but rather an appropriate "parameter" or value for them is typed. For example, acceptable values for <MESSAGE TYPE> in the SIGMA message service are AUTODIN, Memo, Note or Informal. The acceptable values for <SECURITY> are Unclassified, Confidential, Secret or Top Secret, as expected. These will be explained in detail below. Whenever a term appears in angle brackets, then, the user must enter an appropriate parameter value.

Sometimes parentheses appear instead of angle brackets. The parentheses mean that the term enclosed does not have to be entered, but only appears for user readability. For example:

QUIT (LESSON)

The above instruction exits from a Lesson. The user must enter QUIT. He may optionally enter the word LESSON. The service behaves the same way in either case.

Note that any instruction that is typed in must be completed by pressing the EXECUTE key.

Some instructions are represented by Function keys (in particular, by Instruction keys). These are not typed in, but are entered by pressing the appropriate key. Such Key Instructions are shown throughout the documentation as hyphenated instead of separate words, and are surrounded by double exclamation points. For example:

!!FILE-DIRECTORY!!

The above Instruction key shows all files belonging to the user.

SERVICE FEATURES

Before embarking upon the details of the documentation, there are several special features that need to be described.

To log on to the service, you must fill in your logon name (which may be an office code), your logon password, and maximum security level at which you plan to work. If you are logging on as an office code, you must also fill in your personal identity (name) and personal password. To fill in these items on the screen, use the cursor movement, editing keys, and then press the EXECUTE key. When the user starts SIGMA, the required logon information must be filled in on the screen and EXECUTEd within a few minutes, or the terminal is automatically cleared.

If the user attempts to run SIGMA from a terminal which is not authorized for the purpose, or enters a password incorrectly too many times during the Logon process, SIGMA will lock the terminal. This is done for security reasons. Once this happens, the user must contact the System Control Officer (referred to as the SCO) or someone else who knows how to deal with locked SIGMA terminals.

Immediately after Logging On to SIGMA, the screen will show News in the View window (half-intensity). This will describe any new features or procedures relevant to users of SIGMA. Once the user has read this News, the View window can be cleared with the CLEAR-VIEW key.

When a user wishes to work with messages directed to him personally, rather than to his office code, he may log on to the service with his own name and password, or he may log on as an office code, then IDENTIFY as his own name. Either case gives him access to his own Pending file. When he logs on as himself, the file names Pending and Mypending will access the same file. The name Mypending is for the purpose of allowing the user to see items addressed to him personally while he is logged on in another role, i.e., an office code.

As another protection mechanism, users are initially considered Novices by SIGMA, and are required to CONFIRM all Typed instructions. Upon typing an instruction and pressing the EXECUTE key, the user will be prompted by the service to confirm the instruction. The instruction will be redisplayed on the screen as it is understood by the service, that is, abbreviations will be expanded and spelling errors corrected. This offers the user positive feedback that the service understands the instruction he entered and will perform it upon confirmation. The user confirms by pressing EXECUTE again.

This Manual, the on-line HELP, and all Lessons and Exercises describe the service from the point of view of a Novice user. This is most apparent in terms of confirmations required for instructions.

Once a user becomes familiar with SIGMA and no longer wants the safety of confirming each instruction, he can ask the System Control Officer (SCO) to make him an Intermediate user. This will make SIGMA ask for confirmation only for instructions which warrant it.

Similar to confirmation is ACKNOWLEDGMENT. Because of security and other requirements, the user is sometimes required to acknowledge an act about to be performed by the service. In each case, the service's prompt should be self-explanatory and the user must acknowledge by pressing the appropriate key (YES or NO). An example of use of this acknowledgment is the page-by-page acknowledgment required upon changing the classification of a text object. An acknowledgment is also required when you want to delete multiple entries in a file, and in certain other cases.

FIGURE 1-a

Autodin Message in Preparation (Before Chop)

```
Ref Id: TOMW 220018Z MAY 79 (TOMW's Version)
CONFIDENTIAL [Autodin - Preparation]
Briefing Memo:
        Precedence: R(ROUTINE)
                From:   CINCPAC HONOLULU HI
                To:
                Info:
                Exempt:

                C O N F I D E N T I A L
        Subj:
        Ref:
        Text:
        Downgrade Instructions: DECL
Chop:
Release:
Originating office: J3
Distribution: TOMW
 ---END OF MESSAGE---
```

Autodin Message in Preparation (Chopped)

```
Ref Id: TOMW 220018Z MAY 79 (J301's Version)
CONFIDENTIAL [Autodin - Preparation]
Briefing Memo: This field will not be included in the final released message.
        Precedence: R(ROUTINE)
                From:   CINCPAC HONOLULU HI
                To:     JCS WASH DC
                Info:   CINCPAC HONOLULU HI
                        NAVSTA SUBIC BAY RP
                Exempt:

                C O N F I D E N T I A L
        Subj:   FORTHCOMING EXERCISE
        Ref:    A. MY 04001Z JAN 79
        Text:   THE UPCOMING EXERCISE NEEDS FURTHER DEFINITION.  THE GOALS
                SPECIFIED SO FAR DO NOT DETERMINE THE SCOPE OF THE OPERATION.
        Downgrade Instructions: DECL
Chop:
Release:
Originating office: J3
Distribution: TOMW
TOMW's Chop:  J301:chop_YES by TOMW, J38:not_sent, RICHARDSON:not_read,
        DDO:not_sent
TOMW's Release:  J3:not_sent
 ---END OF MESSAGE---
```

FIGURE 1-b

Autodin Transmitted Message

```
Ref Id: SEQ 2100585
CONFIDENTIAL [Autodin - Transmitted]  PSN:752348
R(ROUTINE)
221647Z MAY 79
From:   CINCPAC HONOLULU HI
To:      JCS WASH DC
Info:    CINCPAC HONOLULU HI
         NAVSTA SUBIC BAY RP
Orig:   J3
Internal: MME TOMW
Text:
C O N F I D E N T I A L
SUBJ: FORTHCOMING EXERCISE
A. MY 040012 JAN 79
THE UPCOMING EXERCISE NEEDS FURTHER DEFINITION.  THE GOALS
SPECIFIED SO FAR DO NOT DETERMINE THE SCOPE OF THE OPERATION.
DECL
NNNN
 ---END OF MESSAGE---
```

FIGURE 2-a

Memorandum in Preparation (Before Chop)

Directorate/Memo/Memo Number:
Date:

SECRET

MEMORANDUM

From: TOMW
To:

Subj:

Ref:

Encl:

Text:

Signature block:

SECRET

Copy to:

Orig:
Typed by:

Briefing Memo:
Chop:
Release:
Ref Id: TOMW 212344Z MAY 79 (TOMW's Version)
SECRET (Memorandum - Preparation)
---END OF MESSAGE---

FIGURE 2-b

Memorandum in Preparation (Chopped)

Directorate/Memo/Memo Number: 821
Date: 21 MAY 79

SECRET

MEMORANDUM

From: TOMW
To: J3

Subj: Test of Coordination of memos

Ref:

Encl:

Text:
Memos, as well as Autodin messages, may be coordinated using SIGMA.

Signature block: Tom Wisniewski
Civilian
Staff Member

SECRET

Copy to:
DDO

Orig: TOMW
Typed by: JBW

Briefing Memo: I want your advice on who else should get this memo.
Chop: RICHARDSON STOTZ TUGENDER
Release:
Ref Id: TOMW 212344Z MAY 79 (TOMW's Version)
SECRET (Memorandum - Preparation)
---END OF MESSAGE---

FIGURE 2-c

Transmitted Memorandum

821
21 MAY 79

SECRET

MEMORANDUM

From: TOMW
To: J3

Subj: Test of Coordination of memos

Memos, as well as Autodin messages, may be coordinated using SIGMA.

Tom Wisniewski
Civilian
Staff Member

SECRET

Copy to:
DDO

Orig: TOMW
Typed by:JBW

Releaser: TOMW
Ref Id: TOMW 221639Z MAY 79
---END OF MESSAGE---

FIGURE 3-a

Informal Note in Preparation

```
Ref Id: TOMW 220838Z MAY 79
CONFIDENTIAL (Note - Preparation)
From:   TOMW
To:
CC:
Subj:
Ref:
Text:

 ---END OF MESSAGE---
```

FIGURE 3-b

Transmitted Informal Note

```
Ref Id: TOMW 220843Z MAY 79
CONFIDENTIAL (Note - Transmitted)
From:    TOMW
To:      J301
CC:
Subj:    Meeting tomorrow
Ref:
Text:
Can we move the meeting on SIGMA tomorrow from morning to afternoon?
Releaser: TOMW
 ---END OF MESSAGE---
```

FIGURE 4

Pending File Display

```
File: PENDING Security: SSSS Length: 15
 1 O UU Auto 042222Z DEC 77 From: JCS WASHINGTON DC
        Incoming   Cog: J3
        Subject: AIRCRAFT HIJACKING
 2 R UU Auto 010024Z DEC 77 From: CINCPAC REP PHILIPPINES SUBIC BAY RP
        Incoming   Act: J03
        Subject: INFORMATION ON GAO ACTIVITY
 3   SS Memo STOTZ 020011Z DEC 77 From: stotz
        Incoming
        Subject: Check out a message for me.
 4   UU Note AMES 020333Z MAR 79 From: AMES
        Incoming   Subject: Action on Notes
 5 R UU Auto STOTZ 280038Z JAN 79
        for_Chop   By: TOMW
        Subject: Exercise Procedures
 6 R UU Auto STOTZ 280040Z JAN 79
        for_Releas By: TOMW
        Subject: Elegant Eagle
 7 R SS Auto 082359Z OCT 76 From: NISC WASHINGTON DC
        for_Info   By: STOTZ Act: J2 J3(by STOTZ) TOMW(by STOTZ)
        Subject:  NOFORN/WNINTEL //N03822// SECTION 01 OF 02
 8 R SS Auto RICHARDSON 082358Z OCT 78
        File_copy  By: TOMW
        Subject: RADDR FBIS BANGKOK//120852Z DEC 77
 9   SS Memo TOMW 070027Z MAR 79 CID: TOMW 070020Z MAR 79
        Back_copy
        Subject: Handling of last week's exercise
10 R UU Auto 040130Z DEC 77 From: FBIS HONG KONG
        for_Action Act: J2 TOMW(by J301)
        Subject:  H BBC RUMJ
11 R UU Auto 041647Z DEC 77 From: FBIS OKINAWA JA
        for_Action Act: J2 J301(by J301)
        Subject:  BBC
12 P UU Auto 040555Z DEC 77 From: FBIS BANGKOK
        Retrieved  Act: J2
        Subject:  BBC RUMT
13   SS      AAA 230648Z MAY 79
        ERROR      XMIT-FAIL: Your message had no TO list.
14 R SS Auto TOMW 232332Z MAY 79
        Chop(Y)-Ed By: J301
        Subject: Exercise Procedures
15   CC Memo TOMW 232338Z MAY 79
        Chop(N)    By: J301
        Subject: Review Meeting
 ---END OF FILE---
```

III. MESSAGE FORMATS

The message service supports three forms of communication: formal AUTODIN record messages, internal record messages (Memoranda), and Informal Notes that are unofficial.

AUTODIN MESSAGES

SIGMA Message Form for Message Preparation

The basic SIGMA Message form for messages in preparation (Figure 1-a) is based on the DD-173 form extended to utilize features of the on-line service. It provides features and insures policy restrictions associated with DD-173. It provides, in some instances, more generality and it reduces some of the restrictions arising from the mechanics of typing the message on printed form. Messages are shown by SIGMA with a black-on-white "---END OF MESSAGE---" mark at the end. (Other objects have similar <End of Object> marks at their ends.)

The basic fields shown in Figure 1-a always appear on the message form, similar to the preprinted fields of the present DD-173 form. Note that fields such as Info are included in the SIGMA Message form (though they are not on the DD-173) because the electronic service can automatically expand and contract field sizes to accommodate variable length entries.

Each field of the SIGMA Message Preparation form is described under MESSAGE DRAFTING.

Transmitted Messages

The preceding description applies to the format of the SIGMA Message form as used to prepare messages. When messages are released and transmitted, the recipients view them in a slightly different form. Figure 1-b shows an AUTODIN transmitted message.

The following fields of the SIGMA Message preparation form, used during preparation and coordination, do not appear upon reception: briefing memo, chop list, release, chop status, release status, downgrade instructions, and reviewers' comments fields. Other new fields do appear, such as DTG, Internal Distribution, and a single field designated with one of the three names: Action, Cog or Orig. (Compare forms shown in Figure 1.)

INTERNAL MESSAGES

So far the discussion has addressed messages for, or received from, AUTODIN (via LDMX). The service also supports internal messages between users of the message service. These messages may either be Memoranda or Informal Notes.

Memoranda

A Memorandum is of the same basic message form as the AUTODIN message. It is distinguished from the latter by the second line of the message and by the fact that the addressees that are filled into the From, To, and Copy To fields on the SIGMA Message Preparation form must be local users of the SIGMA service. (See Figure 2.)

Informal Notes

The Informal Note is for informal correspondence. The format of a Note is similar to the SIGMA Memorandum form with the following fields omitted: briefing memo, chop list and release field. Note that the Info header is labeled "CC" on this form. SIGMA does not support coordination of notes. (See Figure 3-a and 3-b.) Informal Notes cannot be Coordinated.

Message ID

For messages in preparation and released Memoranda and Informal Notes, a combination of the drafter's name and the date-time group can be used to refer to the message uniquely. For received AUTODIN messages, a unique SEQ number is assigned. Users can access an AUTODIN by this SEQ number directly. Users may also access an AUTODIN message by its DTG by using a special file which SIGMA keeps, called a Datefile (see File System below).

IV. FILE SYSTEM

The message service features an electronic filing system that is conceptually similar to a filing cabinet in that you can have any number of files, and each of them can contain up to 6,000 messages. An example is your Pending file. These can include messages for information or action, drafts for chop, messages which you have sent, and so forth. These entries representing your messages constitute your 'Pending file.

These entries represent your messages in that they are only pointers to the messages, not the messages themselves. For example, you can manipulate entries by filing them, moving them from the Pending file to other files, delete them from the Pending file as you work with them, etc., but you cannot in fact delete a message itself.

In addition to your Pending file, you can have any number of personal files that you want to define. You may organize according to subject, or organize according to action, info, classification, incoming, outgoing, or whatever you desire -- any arrangement you like. Additionally, there can be a number of files that may be accessed by you and others as well, such as readboards.

The kinds of things that you can do with these files are, of course, to put messages into them or take messages out for display.

You may create files for storing messages that you've received or sent, and these files may be named and organized to suit your needs. You also have the ability to file messages into the files at your own discretion, based on your own filing criteria. Files may be named in any way you wish, except that multi-word names are separated by an underline rather than spaces, e.g., Merchant_Ships_Jan_77.

With these files, you may see only a part or a subset of the file that is of interest to you. For example, let's say that you have a personal file labeled SECRET containing your SECRET messages, and you only want to look at those that were originated by PACAF. You could extract only those entries, or the messages themselves, of interest to you, instead of looking at the entire file. We have several ways of specifying a particular message in a file of perhaps several hundred. We'll cover that later.

When you wish to see the contents of a file, SIGMA will display summaries of the messages the file contains. These are called file entries (see Figure 4). These entries show fields or parts of fields extracted to provide this summary information. An entry contains entry number, precedence, type, classification, date-time group, originator, and subject. Entry numbers adjacent to each entry provide an easy way to reference the message corresponding to that entry. The action indicator shows a user's duties with respect to the message, e.g., action, to be chopped, transmitted successfully or not, etc.

There may be a Current Entry in a file. This is the entry the user last looked at or referenced. It is marked by having its number shown as white-on-black instead of black-on-white. Several instructions and operations can be performed with reference to the Current Entry or the Next Entry (the one immediately following the Current Entry).

Access control to files is restricted by the security level of the file. The security level may be one of four classifications: Unclassified, Confidential, Secret, or Top Secret (see Figure 4). The messages in a file may be of lower classification than the classification of the file, but may not be higher.

A user may annotate a file entry by using the COMMENT instruction. The comment may be of any length since the service automatically expands the comment area. Comments may be specified as being private to the commentor, personal for some named user, or public.

A user may also highlight an entry in a file. This causes the Subject line of the selected entry to be shown in inverse video. Highlighting can be made private to the user, personal for another user or public.

Files should be kept down to manageable size. When a file grows larger than 5000 entries, SIGMA will show only the first 5000 entries, and will alert you to the fact that you are not seeing the most recent entries (those after entry 5000). This is done to warn you that the file should be reduced in size or made into more than one file. If the file is allowed to grow indefinitely beyond 5000 entries, new entries may eventually be lost.

Entries in Files are normally kept in the order in which they are entered into the file; most recent is last. However, you may have SIGMA "sort" the file into Date Time Group (DTG) order; most recent DTG is last.

SPECIAL FILES

The Pending File

Associated with each office code are Pending files which contain entries for messages and drafts awaiting attention. These Pending files are the basic daily working files. Entries in these files include received AUTODIN, Memoranda, Informal Notes, messages for Action, for Chop, etc. Messages in preparation are automatically filed in the Pending file when they are created. The user can explicitly file them in some other file.

Each entry in the Pending file is actually a pointer to the message, that is, it represents a message. These entries each include the following information (common to all files): entry number, precedence, type (For Chop, etc.), classification, date-time group, originator and subject. To see the message itself, you would use one of the DISPLAY instructions, such as

DISPLAY ENTRY <NUMBER>

If a user has his Pending file open, most new entries are dynamically updated as they arrive. If the end of the file is being displayed, they are appended on screen. These new entries may be treated like any others (i.e., they may be deleted, commented and the message they reference displayed). If another user does a FILE or COMMENT to the user's Pending file, these will NOT appear. The user must close (!!FINISH!! or ABORT) and reopen (DISPLAY PENDING) the file to see them.

Pending files are unique in that they are divided by classification. Top Secret entries are in a separate grouping, apart from Secret and below.

To see the Pending file, use the instruction

DISPLAY FILE PENDING <SECURITY>

Mypending File

This file is similar to the Pending file in that it contains a list of entries awaiting the user's attention. This is the name given to the user's personal Pending file when he is logged on as an office code. If you log on as J3 and IDENTIFY as Smith, you will see messages sent to J3 in the Pending file, and messages to Smith in the file named Mypending. When you log on as Smith, you only have Smith's Pending file (which you can access with either of the names Pending or Mypending).

To see your own personal Pending file when you have logged on as an office code, use the instruction

DISPLAY FILE MYPENDING

The ACTION_LOG File

Whenever action is assigned to a message, it is recorded in the ACTION_LOG. In order for each user to have access to it, he must issue the following instruction:

GET FILE ACTION_LOG (FROM) <USER ID> e.g.,

GET FILE ACTION_LOG FROM J3

This instruction transfers the office Action Log to the user as a dynamically shared file which will be kept up automatically by the service as each new action assignment is made.

If you are assigning action to messages, the assignment will not be recorded unless you have an Action Log in your files. You may CREATE an Action Log if you have the authority, or you must GET the already existing Action Log from the appropriate office.

Readboards

Readboards are basically the same type of file as the ACTION_LOG. Any user who has authority to access the office readboard need only use the instruction

GET FILE <EXISTING FILE NAME> (FROM) <USER ID> e.g.,

GET FILE Daily_Readboard FROM J301

where the <EXISTING FILE NAME> is the name of the readboard and the <USER ID> is the appropriate office code.

Datefiles

Datefiles are used for the retrieval of messages by DTG (date time group). Usually a message will come to the attention of a user by its presence in a file he uses (Pending file, Readboard or personal file) and will be tagged by its entry number. Occasionally a message will come to a user's attention by some other means (e.g., referenced in another message), in which case the user's only identification is the originator and DTG of the message.

To retrieve such a message the user will first open a special file called a DATEFILE for the date of the given DTG. These files have entries for all messages with DTG's for that date. (Note that this is not the same as a log of all messages received on that date, since the DTG of a message may not correspond to the date it is sent or received.)

A Datefile is opened at a given security level and will show entries at and below that level. That is, if opened at SECRET, all UNCLASSIFIED, CLASSIFIED and SECRET entries will be shown. For example, if the user wants to find a SECRET message with a DTG of 281011Z MAR 77, he would open the appropriate Datefile by executing a DISPLAY DATEFILE instruction, such as

DISPLAY DATEFILE 28 MAR 77 SECRET.

This will display a synopsis of all messages with DTG's within that date (up to SECRET). The user then performs a

RESTRICT FROM <ORIGINATOR> AND DTG <DTG>

such as

RESTRICT FROM "JCS" AND DTG 281011Z MAR 77.

This will show him the entry number for the desired message, which he may then display, file or whatever. The user must enter the year, month, and full six-digit DTG in doing this RESTRICT, even though the file is already limited to entries of a single day.

If the user is not sure of the DTG of the message, he may do a

RESTRICT FROM <ORIGINATOR>

and examine all the messages received from that source. Presumably he can select the message of interest from the Subject lines shown. If the user is not sure of the date, he may have to examine more than one file to get the message of interest.

V. ALERTS

The Alert facility calls a user's attention to new messages as they arrive. The user must specify what messages he wants by defining a standard SELECTOR with the special reserved name ALERT_SELECTOR. This selector defines the criteria which an incoming entry must pass to be considered an ALERT. As entries arrive for a logged-on user's Pending file, they are compared against the ALERT_SELECTOR automatically.

The number of ALERTs that have arrived is shown on the Flash line next to the new Pending file notification. Thus the flash line might say "Alert:1 Pend:7." Each time an ALERT arrives the bell is sounded.

An Alert List looks like a SIGMA File except there are two blank inverted video spaces where the entry number normally appears. Each Alert that has arrived shows as an entry in the Alert List. Thus the Alert List looks like the contents of the Pending file that has had a RESTRICT with ALERT_SELECTOR AND RECENT applied to it. The user may do a !!HERE!! on an Alert List entry and a !!DISPLAY_ENTRY!!, which will cause the referenced message to be displayed. The user may also execute the following instructions in conjunction with a !!HERE!! on the Alert List: !!VIEW ENTRY!!, !!COPY ENTRY!!, !!REPLY ENTRY!!, !!PRINT ENTRY!!, FINDSTRING, ACTION, or FORWARD. The user may not do any instructions that require the actual Pending file or the concept of a "Current Entry" such as !! DELETE ENTRY!!, !!DISPLAY NEXT!!, !!VIEW NEXT!!, ROUTE, FILE, COMMENT, or KEYWORD. AUGMENT, RESTRICT, BACKUP and FIND ENTRY do not have meaning for an Alert List.

If the Alert List is being Viewed when a new Alert arrives, the Alert List is updated dynamically and the user will see the new entry added at the bottom.

VI. MESSAGE DRAFTING

INITIATING THE MESSAGE DRAFT

The drafter requests a preformatted message form in order to input a message draft on-line to the service. The CREATE instruction described later shows how to request these message forms.

After requesting the appropriate form, the drafter fills in the various fields by typing in their respective values.

Outgoing AUTODIN Messages

The SIGMA Message form (see Figure 1-a) is used for record traffic. This form contains fields that are to be filled in by the user using the editing capabilities of the service. The following discusses each field on the message form and how it is to be filled in.

Classification

When requesting a blank message form, the user must indicate the classification level for the message. The security level chosen is then shown in the field labeled Header.

Reference ID

The Reference ID field is the identified organization code (or personal name if no organization code has been given) of the person logged on to the service requesting the message draft and a date-time group assigned by the service. This is also called the Message ID and may be used for message retrieval.

Briefing Memo

The Briefing Memo, an optional field, provides space for an explanation of the message. It is filled in by the drafter with data specifically for reviewers and the releaser. That is, it does not appear on the message as delivered to action and information addressees. It is similar to the message body in that it contains unstructured text.

From

For a message for AUTODIN, the From field is automatically filled in by the service with CINCPAC HONOLULU HI.

To, Info, and Exempt Lines

For outgoing AUTODIN messages, Plain Language Address (PLA) codes are used. Exempt refers to exemptions from any Address Indicator Groups (AIGs) appearing in the To

and Info fields. A carriage return separates the PLA's (50 character length restrictions apply). Five or more spaces are used as the first characters on subsequent lines to indicate that a PLA spans more than one line.

Precedence

The Precedence field is automatically filled in with ROUTINE. This field can be changed by the user. Acceptable values for precedence are R (Routine), P (Priority), O (Immediate), Z (Flash) and Y (Flash Override). The Precedence field can take two entries: the first for the Action addressees, the second precedence for those in the Info field.

Subject

The contents of this field may be any text. On the outgoing message sent to LDMX, the Subject appears as the second line of the Text field of the message. Although the service does not enforce it, this is usually a single line of text.

References

This field appears after the subject on the Preparation AUTODIN message. When composing a message using SIGMA, references may be typed in or copied from another message. References may be to letters, telephone conversations, etc., as well as to other messages. Thus any arbitrary text is acceptable. On the Outgoing message sent via LDMX, the Reference field appears after the Subject, inside the Text field of the message.

Text

The message text is entered in the field labeled TEXT. The service dynamically expands the space between TEXT and DOWNGRADE INSTRUCTIONS to accommodate the desired message size. The service does not examine the text.

Downgrade Instructions

This field follows the text of classified messages. The default value is GDS, which is inserted by the service. This may be altered by the user. The service does not check the validity of what the user enters.

Chop

The Chop field on a SIGMA Message in preparation is filled in with the names of people (or offices) with whom you want this message coordinated. When the message is coordinated, copies of the message are sent in parallel to all names on the list that are marked with !!HERE!! (see Terminal Keys below). Each recipient will get a "For_Chop" entry in his Pending file. Office codes or individual names may be entered.

Drafter Chop

When a user receives a message in preparation which requires his chop, the form allows him to fill in his own chop field to indicate other users with whom he wants the message coordinated. The original chop field on the drafter's version of the message is then shown as Drafter Chop.

Chop Status

Once a message has been coordinated, the Chop field is followed by a Chop Status field which shows what action the various reviewers have taken so far on the message.

Release

This field indicates the intended releaser of the message. In fact some other authorized reviewer may choose to release the message first. This user will get a "For Releas" entry in his Pending file. In order for him to be sent the message, his name must be marked with a !!HERE!! and !!COORDINATE!! key pushed.

Drafter Release

For a reviewer other than the drafter himself, the Release field is the reviewer's suggestion as to who should release the message. The drafter's suggestion is then shown in the Drafter Release field.

Release Status

This field shows what action the designated releaser has taken on the message, just as the Chop Status field shows what action other reviewers have taken.

Originating Office

This field indicates the office code that will be shown as the originator of the message in LDMX (e.g., J3, DDO, JRC).

OTHER FEATURES

Comments on Message Fields

With the COMMENT instruction, a drafter or reviewer may append a comment to any message field or text paragraph. Thus a drafter or reviewer may conveniently key his remarks to specific fields of the message. Each comment field is dynamically expanded to allow the user to enter a variable length comment.

MEMORANDA

Memoranda are messages that are addressed to people or organization codes within CINCPAC, but that are formal in that they represent official business and are recorded for later evidence. This form of communication does not have a counterpart in current manual message operations. (See Figure 2.)

These formal Memoranda are prepared and received using a form very similar to the AUTODIN form (Figure 1-a and 1-b) described above. The same fields apply for Memoranda (Figure 2) with the following exceptions.

From

The From field for internal traffic is the organization code of the officer authorizing the message.

To, Copy To

These fields should be names or organization codes of personnel within the command. They can be people or codes unknown to the message service, in which case they will not be able to access the message through the service. Presumably the author will deliver printed copies to these addressees manually.

Exempt

This field has no meaning for an internal message since there are no AIGs. Therefore, it is not provided by the service.

Originating Office, Downgrade Instructions

These fields have no meaning for an internal message.

Release

Memoranda are formal messages that allow Coordination and Release like AUTODIN Messages. However, Memoranda are purely internal messages, which are NOT released to AUTODIN. For these messages, the Release field contains the name of whoever has the authority to send the Memorandum internally.

INFORMAL NOTES

These messages are for off-the-record communication, comparable to the telephone.

The message form used for Informal Notes (Figure 3-a and 3-b) is somewhat different from the SIGMA Memorandum form. The fields are treated the same as for Memoranda except as described below.

CC

This "carbon copy" field is the equivalent of the Info field on the SIGMA Memorandum form.

Release, Release Status

These fields are absent for Informal Notes, since they are not formally released at all. The drafter sends an Informal Note himself.

Chop, Chop Status

The Chop and Chop Status fields do not appear in Informal Notes since one cannot coordinate an Informal Note.

REFERENCES

Except for the first of a series of messages on a given topic, a message is usually in response to a previous message received. Thus, the composer (i.e., the message drafter) uses the earlier referenced messages in the series as the basis for composing the new message. The automated service is useful to the drafter in retrieving references to be displayed, viewed, or printed. Generally, it is expected that users will wish to quickly scan the references on-line to determine which ones are needed in constructing the new message.

USER LISTS

Often a user deals with a list of addressees over and over again. These may be used in the To, Info, review or distribution fields. To eliminate having to retype these names each time they are used, the user may generate them once and store them as TEXT OBJECTS. Then whenever the list is to be inserted the user merely puts the list into the appropriate field. (See the PUT instruction.)

COORDINATION

When the drafter sends a draft message out for coordination, copies of the message are sent in parallel to all users on the chop list that are marked with a !!HERE!!.

TRANSMISSION

The message is actually sent by execution of the RELEASE instruction. A message of each type (AUTODIN, Memo, or Informal) may only be released by a user whose name appears on a special list of authorized releasers for messages of that type. This authorization can be set or changed by the SSO. The message service will not accept the message release instruction if the user is not on this list.

CORRECTING ERRORS

If you should ask for the wrong message form, e.g., Memo instead of AUTODIN, use the ABORT instruction to clear the screen, and ask for the correct form. If you are using the correct form, but decide you want to start over again, use the ABORT instruction. Other errors can be corrected by editing with the cursor. All corrections or changes to a message should be made before it is sent for coordination or released. Once a message has been completed, it becomes a part of the SIGMA service, and cannot be deleted.

Assume, however, you composed a message and filed it before deciding to coordinate or release it. Suppose for some reason the need for the message no longer exists. The only way to "delete" this unsent, unneeded message is to delete all entries in your files that point to it.

VII. MESSAGE REVIEW and COORDINATION

When a new Memo or AUTODIN message is created, several fields appear which are used only in the preparation phase for the Coordination or Chop process. These fields do NOT get sent with the message when it is finally released. They include the Chop list, Release field and the Briefing Memo. (See Figures 1-a and 2-a.)

The Chop List

When you draft a message, you fill in the Chop field with the names of those people with whom you want to coordinate the message.. Each person on the Chop list is a Reviewer who will be able to suggest changes and make comments on your proposed message and then Chop either Yes or No on it. (If the Release field contains the name of the intended releaser for the message, he is also considered a Reviewer.) In order to send the message out for Chop to one or more of these people, you mark their name with a !!HERE!! and then push the COORDINATE Instruction key. The COORDINATE key requires acknowledgment. This puts a For_Chop entry in the Pending file of each Reviewer indicated to tell him that his Chop is requested. If the name in the Release field is marked, that user will receive a For_Release entry in his Pending file.

Each Reviewer can mark up and comment on his copy. When he is done, he Chops on it by pushing CHOP_YES or CHOP_NO to return it to the Drafter. The copying and delivery of messages in Coordination is done automatically by SIGMA in response to the COORDINATE and CHOP keys.

When a Reviewer Chops (either Yes or No) on the message, a Chopped entry is automatically put in the Drafter's Pending file, signalling that this Reviewer has Chopped. In addition, the status of each Reviewer is shown in the Chop Status field of the message. This field shows the name of each Reviewer, the status of his action on the message (Not Sent, Not Read, Read, Chopped Yes, or Chopped No), who was the Executor of the Chop (name of the person logged in as the office code that executed the chop), and whether the Reviewer has made any edits to his version of the message.

When a message is sent for Chop, new versions are created for each Reviewer. These are still referred to as the same message, but they allow the Reviewer to make changes and add comments to his own version of the message. The Drafter can compare his version with a Reviewer's returned version after it has been Chopped. The Drafter can then decide whether or not to incorporate suggested changes into his own version. The Drafter can look at a Reviewer's Chopped version of the message with the typed VIEW instruction VIEW VERSION <USER ID>.

The Reviewer's version does not belong to the Drafter, so he cannot edit it himself. But the Drafter can VIEW other versions while editing his own version in the Display window and can use the COPY TEXT instructions to copy parts of the Reviewer's version into his own.

A Reviewer can also use the VIEW VERSION instruction to see someone else's version of the message, for example the Drafter's. This would show him any comments attached to the Drafter's version.

Whenever a coordinator DISPLAYs a message for which a "For_Chop" or "For_Releas" citation has been newly received, he will get, as his version, an exact copy of the version sent to him (less comments). The Header line of the message will identify who sent him this message. Thereafter his version will contain whatever editing changes he makes. When he does a !!CHOP!!, !!COORDINATE!!, or !!FINISH!!, his version becomes public (i.e., may be seen by other coordinators and the drafter).

If after he has done a !!FINISH!!, he is sent another citation to the same message, his version is again copied from the version sent to him, but his previous version is maintained and may be accessed by him by means of the VIEW VERSION <User ID> instruction where he specifies his own ID or leaves User ID blank. His "old" version is lost whenver he does a !!CHOP!! or !!FINISH!! of his new version. VIEW VERSION always shows the last version for which he has done a !!CHOP!! or !!FINISH!!.

Message Review

You are notified that there is a message requiring your Chop by an entry in your Pending file (or in the Pending file of the Office code for which you are serving). This entry is identified as For_Chop and allows you to see your own copy (version) of the message the Drafter has sent for Coordination. At this point your status on the message is "Not Read". This means you have been sent a copy, but have not yet displayed it. If you display this entry, you will see your own version of the message, and your status will be changed to "Read." Users on the Chop list who were not sent the message yet will show a status of "Not Sent."

As a Reviewer, you would then read the message and make any changes you feel are necessary. You can do this by editing your own copy to make it read the way you feel it should, or by putting comments on any of the fields of the message. When you have completed editing the message, you indicate the disposition you wish to make by pushing either the CHOP_YES or CHOP_NO key. This will cause a "chopped" entry to appear in the Drafter's Pending file.

When you are Coordinated on a message, your version is initially almost identical to the Drafter's version. However, any comments the Drafter has attached to fields of his version are not shown in your version. Also, your own Chop and Release fields are shown as initially blank, while the Drafter's Chop and Release fields are shown as "<drafter> Chop" and "<drafter> Release". To see if the Drafter has comments on any of his message fields which require your attention, you would view his version of the message.

In order to see the Drafter's version of the message, you must first have displayed your own version of the message. You can then use the Typed instruction VIEW (VERSION) <Drafter> e.g., VIEW VERSION J301 if J301 was the Drafter. This would show you any comments you have access to on the Drafter's version, and allow you to compare his original with your own version after you have changed it. Similarly, you can VIEW the version of any other Reviewer of the message. If J312 is also on the Chop list of the message, you can VIEW his version as soon as he has looked at it himself.

The Chop list of people with whom the Drafter is Coordinating this message is shown in your own version as the Drafter Chop field. If you feel the message should be

Coordinated with someone whose name does not appear on the Drafter's Chop list, you can place a comment on the Drafter Chop field explaining to the Drafter why you are making this suggestion. Similarly, if you felt someone else should be designated as the eventual Releaser for the message, you could suggest this by editing your own Release field.

As a Reviewer, you may also want to further Coordinate the message yourself. This is done by filling in the Chop list on your own version. This Chop list is used just as if you were a Drafter and were Coordinating the message yourself. You can mark those names on your Chop List which you want sent copies and push the COORDINATE Instruction key to send the message. When your Reviewers Chop on the message, you will be notified by Chopped entries in your Pending file. You can VIEW their versions, incorporate their suggestions and comments in your own, and Chop yourself on the message.

In other words, as a Reviewer, you can do all the things the Drafter does with the message. The difference is that when you are done, you will most likely Chop on the message to send the result of your work back to the Drafter. When the Drafter himself is done, he will either Coordinate the message further, or Coordinate it with the Releaser (if he has not already done so) to get it released.

Multiple Coordination Cycles

The Drafter gets back entries when the message is Chopped by various Reviewers. He may Coordinate with some other set of names, or he may send the message back for Coordination several times to the same people. For instance, if he has incorporated changes and suggestions which make the message substantially different, he may feel the same people should Chop again. He may also decide to Coordinate with other users on the Chop List.

The Chop Status and Release Status fields keep track of who has Chopped on the message already. A Reviewer can Chop Yes or No several times in succession on the same message, if he has reason to change his mind. Each Chop will result in a new Chopped entry being sent back to the Drafter, and the Chop Status field of the message will reflect the last Chop for each Reviewer. If the drafter has edited the message since a user has chopped, that user's status will show "Chopped (Y/N) Old."

VIII. THE TERMINAL SCREEN

The terminal screen has five lines at the top: the Alert Line, the Status Line, a Feedback Line, and two Instruction Lines.

The ALERT LINE is the top line of the screen. It has changing information on it including the SIGMA Release number, the service's load average (a figure indicating how much work is being done by the computer), the day of the week, the date, local time and Zulu (GMT) time. It also notifies you of arriving messages in your Pending file and in your Alert list. This appears as "Pending:two" if there are two new entries in your Pending file, as "Mypending:1" if there is a new entry in your Mypending file, or "Alert:1" if there is an entry in your Alert list.

The STATUS LINE is the second line on the screen. It contains the following information: the security level at which you log on, and the security level, type and name of any open object (up to three maximum allowed). An open object is a message, file, or piece of text you are working on. You can have at most one of each kind of object open at once. A sample line might look like this:

J3(S) <F:S>name-of-file(153/345) <M:C>SMITH 010258Z JUN 77

where "J3(S)" indicates your user id (or the office code you used) followed by the security level at which you are logged in (in this case SECRET); "<F:S> name-of-file(153/345)" means that you have opened a file which is SECRET (whose name is shown), in which the current entry is entry number 153 out of 345 entries in the file; you are looking at a CONFIDENTIAL message by SMITH whose DTG is 010258Z JUN 77.

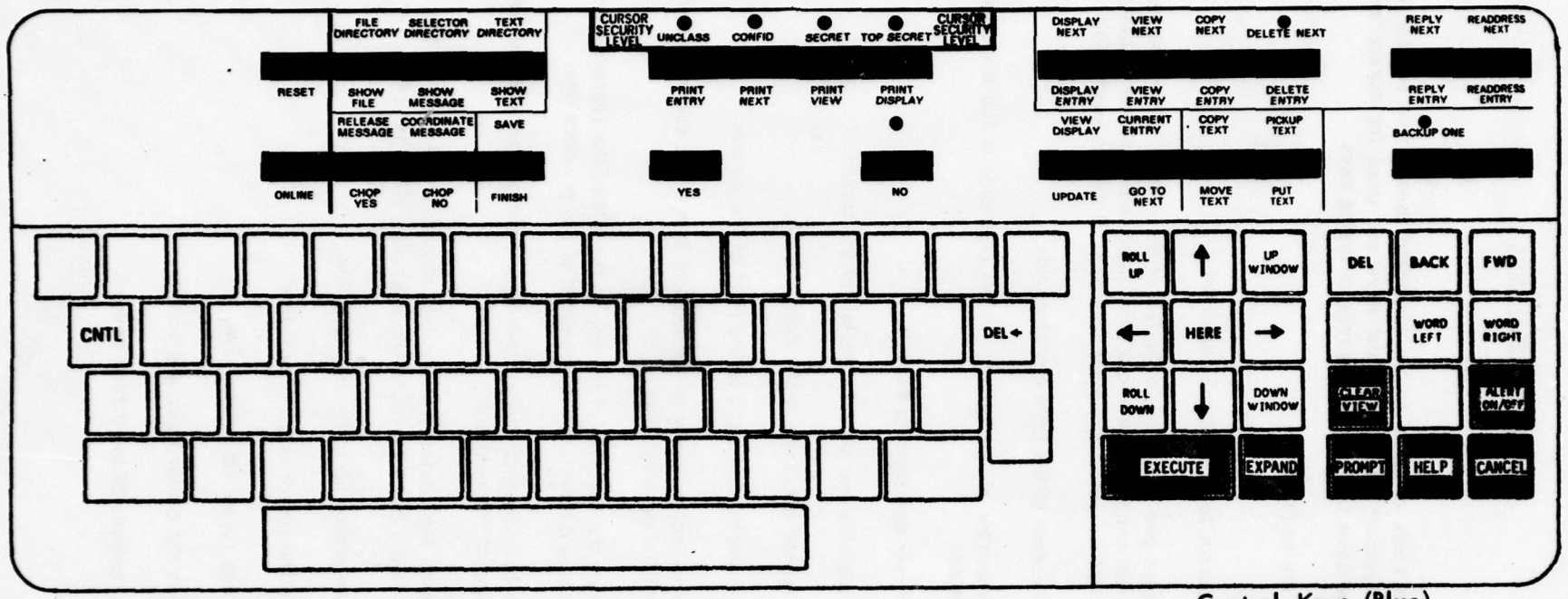
The FEEDBACK LINE begins with two asterisks. It gives you notification from SIGMA telling you what it is doing, such as executing or processing your instruction. You will also get a notification if you make an error in an instruction, or if you are trying to do an operation that is out of sequence.

The Feedback Line may also ask you to confirm or acknowledge instructions. When you are asked to "Confirm" a typed instruction, you press the Execute key. When you are asked to "Acknowledge" an instruction, you must press the YES key (pressing either the NO key or CANCEL will be taken as a negative acknowledgment). These acknowledgments have to do with security operations. There are constant security checks being made by SIGMA, such as monitoring the security level of the contents of the screen, or checking to make sure that higher classified information is not mistakenly placed into a lower classified object.

The INSTRUCTION LINE is where you enter instructions. It begins with two bell-shaped characters.

The remaining portion of the screen is working space. This may be occupied by the DISPLAY WINDOW (display for editing), the VIEW WINDOW, where material you are working on can be placed for reference purposes, or both. The contents of the view window are placed in half intensity to distinguish it from the display window. When both windows are on the screen, the view window is in the lower half. You cannot edit anything in the view window, but you can COPY something out of it. (See the COPY instruction.)

The panel of security classifications on the right side of the screen lights up at the highest level of information present on the screen.



Control Keys (Blue)

THE HP/MME TERMINAL KEYBOARD

1 June 1979 version

IX. TERMINAL KEYS

There are two kinds of keys on the keyboard (beside the regular typing keys): Function keys and Terminal keys. Terminal keys are used for cursor movement and editing. They are the same tan color as the regular typing keys.

Terminal keys are as follows:

Left Arrow - moves cursor to the left one character.

DEL left arrow - deletes one character to the left of the cursor. When the cursor reaches the end of a line, it will jump up one line (at the right) and continue deleting. It will not, however, delete the last remaining character on a line. It can be used as a repeating key.

Right Arrow - moves cursor to the right one character.

DEL - deletes one character at the cursor. If this character is the only one on the line, the entire line is deleted.

Word Left - moves cursor one word to the left.

Word Left (shifted) - deletes one word to the left of the cursor.

Word Right - moves cursor one word to the right.

Word Right (shifted) - deletes one word to the right of the cursor.

Back - moves cursor to the beginning of the current line; if the cursor is at the beginning, it moves to the end of the previous line.

Back (shifted) - deletes the contents of the line to the left of the cursor; if the cursor is at the beginning of the line, it moves to the beginning of the previous line.

FWD - moves cursor to the end of the current line; if the cursor is at the end of a line, it moves to the beginning of the next line.

FWD (shifted) - deletes the contents of the line at and to the right of the cursor; if the cursor is at the end of the line it moves to the beginning of the next line.

Up Arrow - moves the cursor up to the previous line.

Down Arrow - moves the cursor down to next line.

Up Window - moves the cursor up one window.

Down Window - moves the cursor down one window.

Roll Up - scrolls the contents of the screen upward.

Roll Down - scrolls the contents of the screen downward.

Here - places a video marker at the location of the cursor. This key is used with instructions.

X. FUNCTION KEYS

Function keys are divided into three groups: Instruction keys, Acknowledgment keys, and Control keys. Since most of these keys carry two names, a shift is required for the upper name as in regular typing.

Whenever a Function key is pressed, the keyboard locks and the cursor momentarily disappears. You cannot enter anything again until the cursor reappears.

Instruction keys are used to perform certain instructions. All SIGMA documentation, including this Manual, shows such Instruction keys as hyphenated words enclosed by exclamation marks (!!) instead of separate words which would be typed.

Acknowledgment keys are labeled "YES" and "NO". They are used when an instruction requires acknowledgment.

Control keys are the bottom two rows of keys on the right-hand side of the keyboard and are blue or black in color. They are labeled Execute, Expand, Prompt, Help, Cancel, Clear View and Alert On/Off.

EXECUTE is used to complete and confirm Typed instructions.

EXPAND is pressed when the user wants to check to see that an abbreviated form of an instruction will result in the instruction he intends to use. It expands the spelling of each abbreviated word and adds any word to the instruction that is needed.

When you get a message on the Feedback Line telling you SIGMA is unable to interpret your instruction or when you desire more information on the instruction you are entering, press the PROMPT key. The display window will be temporarily rewritten with a list of related or alternate instruction forms which should be sufficient to help you out of your difficulty. The screen is returned to normal when you add to or correct the misunderstood instruction and execute (or cancel) it. If you require further in-depth information, you should use the HELP key.

HELP is pressed to access on-line HELP, which in content is identical to this manual. When the HELP key is pressed, the screen is temporarily rewritten with directions on how to get to the instruction or information you want. As you follow these directions, the contents of the screen will change several times. When you have gotten the information you need, press the CANCEL key, which automatically returns you to the screen as you left it.

CANCEL erases the contents of the Instruction Line.

CLEAR VIEW causes the View window to be cleared.

ALERT ON/OFF is used when the user wants to view his Alert list, which appears in the View window. Depressing it a second time returns the View window to its previous state.

XI. PARAMETER DESCRIPTIONS

Instructions are given to the SIGMA Message Service in two forms: Typed Instructions and Instruction Keys. In the latter case (Instruction Keys) the instruction is generally a simple one requiring no parameters that the user must type. In the case of Typed Instructions, there will generally be some additional parameters, in addition to the Instruction name itself, which the user must type to fully specify what he wants SIGMA to do.

The example below illustrates the form instructions take for the SIGMA Message Service. You will note that some words are presented in parentheses, and some in angle brackets. Parentheses are used to indicate optional words. Angle brackets are used to indicate parameters.

For example, to create an AUTODIN message at the secret level, you would enter the following instruction to the SIGMA message service:

CREATE MESSAGE AUTODIN SECRET

After typing the above instruction, you would press the EXECUTE key and the service would present you with a blank AUTODIN message format (see Figure 1-a) to be filled in.

The general form of the above instruction is presented as:

CREATE MESSAGE <MESSAGE TYPE> <SECURITY>

The terms <MESSAGE TYPE> and <SECURITY> are called "parameters" and are not typed. Rather appropriate values are typed for them. Appropriate values for <MESSAGE TYPE> are AUTODIN, Memo, Note or Informal. Appropriate values for <SECURITY> are Unclassified, Confidential, Secret or Top Secret If you make a mistake in typing and the service can't recognize an appropriate value, a brief message will appear on the Feedback Line and you will be instructed to either correct the erroneous part or use the PROMPT or HELP facilities to guide you to a correct value.

Parameters and their appropriate values for SIGMA instructions are presented below.

<ACCESS SPECIFICATION>

Allowed values: PUBLIC, PRIVATE, <USER ID> (J3 for example)

<CURRENT/NEXT>

Allowed values: Unlike other instruction words shown in angle brackets, CURRENT or NEXT are assumed by certain Instruction keys which act on Current or Next entry depending on whether the Shift key is held down. Current and Next can also be used in the place of an entry number in typed instructions when working with files.

Example: FILE ENTRY CURRENT <EXISTING FILE NAME>

<DATE>

Allowed values: A date can be specified as Month Day Year or Day Month Year as in military format. Months can be abbreviated by three letters. Spaces are used to separate Day Month and Year. YESTERDAY, TODAY and TOMORROW are also accepted dates.

Examples: Aug 12 77 or August 12,1977 or 12 Aug 77

<DTG>

Allowed values: A DTG is a Date Time Group, which is a date and time expressed as a six-digit number for day of the month and time Zulu (GMT) followed by a three-letter abbreviation for the month followed by the year.

Example: 090056Z MAR 78 which is March 9, 1978 at 0056 Zulu.

<ENTRY LIST>

Allowed values: An entry list is a list of one or more entry numbers, separated by spaces or commas. A sequence may be given separated by dashes. The comma and dash may be used in the same entry list. The following list consists of entries: 2,4,5,6,7,10, and 15.

Example: 2,4-7,10 15

If you use the RESTRICT and AUGMENT instructions this way so that the subset of entries you want to ROUTE becomes the restricted display, then you can type the ROUTE instruction with an <Entry List> consisting simply of a "-" which means all the entries in the restricted display.

For ANY Typed instruction which takes an <Entry List> (such as FILE, DELETE, etc.), if you type an <Entry List> consisting of just a hyphen, the instruction will refer to ALL the entries in the current restricted subset of entries. If the entire file is shown (i.e., there is no current restricted subset), then this <Entry List> will refer to ALL entries in the file. For example, the instruction

FILE - Other_File

will file all entries shown in the restricted display into the file Other_File.

Similarly, when you want to DELETE or ROUTE a large number of entries at once and do not want to find and type in all their entry numbers you can Restrict the file to those entries you want and then use an <Entry List> consisting of just a hyphen to refer to all the selected entries.

<EXISTING FILE NAME>

Allowed values: Any file name which the user has previously created. The !!FILE-DIRECTORY!! Instruction key will show you a list of your files.

Example: WEEKLY_ACTIVITY_REPORT_12_SEP_77

<NEW FILE NAME>

Allowed values: A file name which the user has not previously created. Multiple word names must be joined by underscore.

Example: BURNING_CANDY_DEC_77.

<EXISTING TEXT NAME>

Allowed values: Any text name which the user has previously created. The !!TEXT-DIRECTORY!! Instruction key will show you a list of your text objects.

<NEW TEXT NAME>

Allowed values: A text name which the user has not previously created. Multiple word names must be joined by underscores.

Example: AIG_ONE_ONE_THREE_SEVEN.

<EXISTING SELECTOR NAME>

Allowed values: Any selector name which the user has previously created. The !!SELECTOR-DIRECTORY!! Instruction key will show you a list of your selectors.

<NEW SELECTOR NAME>

Allowed values: A selector name which the user has not previously created. Multiple word names must be joined by underscore.

Example: RECENT_MESSAGES.

<HERE>

Allowed values: The HERE key, when pressed, places an inverse video marker at the cursor. This acts as a marker to SIGMA when used with instructions. Unlike other instruction words in angle brackets, this term is not replaced with any parameter, but rather the HERE key is pressed. You cannot type in the word "Here" but must press the HERE key.

The HERE key is used to mark a place on the screen, such as a piece of text, that is to be manipulated in some way, either with a PICKUP, COPY, MOVE, DISPLAY ENTRY, or other instruction. In some cases an instruction may require two markers, telling SIGMA the limits of what you want manipulated, i.e., at the beginning and end of a sentence, list of addressees, text of a message, and so on.

When an Instruction Key requires or allows a <HERE>, the HERE key must be pressed before the Instruction Key. That is, a location must be marked before the Instruction Key is pressed. These Instruction Key forms are shown as <HERE> !!KEY!! to serve as a reminder that the HERE key is pressed first.

Example: See the DISPLAY ENTRY instruction.

<MESSAGE TYPE>

Allowed values: AUTODIN, Memo, Note or Informal.

<Message ID>

Allowed values: For messages in preparation and for released Memos and Informal Notes, the Id consists of the drafter's name followed by the date-time group. For received AUTODIN messages, the Reference-Id is used (this is shown in the message, but NOT in file entries pointing to the message).

<NUMBER> or <ENTRY NUMBER>

Allowed values: Any single number, such as 1 or 57, etc.

<ORIGINATOR>

Allowed values: Either a USER ID, which is the name of a user or office code known to SIGMA (such as J301 or MILLER), or a STRING representing a name not known to SIGMA (such as "FLEWEACEN PEARL HARBOR HI").

<SECURITY>

Allowed values: Unclassified (UUUU), Confidential (CCCC), Secret (SSSS), or TopSecret (TTTT).

<SELECTOR>

Allowed values: A number of the fields and attributes of a message may be used for selection. A selector is a combination of the field or attribute to be used for selection, and the values of that field or attribute that will satisfy the selection criteria. Additionally, selectors may be combined with the words AND, which means

that selected messages must meet both criteria, and the word OR, which means that selected messages need meet only one (or both) criteria. The word NOT may also be used in front of a selection criterion to select those entries which do NOT satisfy the given criterion. Note that some selection criteria may not be applicable to some entries. For example, since a memo has no precedence, it will not be selected either by Restrict Routine or by Restrict Not Routine.

Fields available for Restricting or Augmenting with selectors are:

From <STRING> (e.g., From "J301" or From "FLEWEACEN")
Action <STRING> (e.g., Action "J3")
Subject <STRING> (e.g., Subject "Gale Warning")
Keyword <STRING> (e.g., Keyword "Cambodia")
By <USER ID> (e.g., By J301)
<SECURITY> (e.g., Secret or SSSS)
<PRECEDENCE> (e.g., Routine(R), Priority(P), Immediate(O), Flash(Z), Flashoverride(Y))
<MESSAGE TYPE> (AUTODIN, Memo, Note)
<MESSAGE PHASE> (Preparation or Transmitted)
<RESPONSIBILITY> (Orig, Action, or Cog)
DTG <DTG> (e.g., DTG 021745Z OCT 77)
Before <DTG> (Before and including this DTG, e.g., Before 021745Z OCT 77)
After <DTG> (After and including this DTG, e.g., After 021745Z OCT 77)
Around <DTG> (Within 24 hours of the DTG, e.g., Around 021745Z OCT 77)
<Entry Number> (e.g., 17)
Before <Entry Number> (Before and including this entry, e.g., Before 17)
After <Entry Number> (After and including this entry, e.g., After 17)
Around <Entry Number> (5 entries above and 5 below, e.g., Around 17)
Incoming: a message that has been sent to the user.
Error: an error condition has been detected. A brief description of the error is provided in the entry.
For_Action: a message sent to this user for his action with ACTION.
For_Info: a message sent to this user for his info with FORWARD.
For_Chop: a message sent to this user for his chop with COORDINATE.
For_Releas: a message sent to this user for his release with RELEASE.
Chopped: a message returned as either Chopped(Y) or Chopped(N).
File_Copy: an entry for a message that has been placed in the file.
Back_Copy: a comeback copy of a released message.
Retrieved: a message retrieved from Archive.
Exercise: entries pertaining to Exercise operations. (When implemented.)
Deleted: entries deleted from the open file during that work session.
All deleted entries are destroyed after the file is FINISHED.
Recent: newly arrived entries in a file since it was last opened.
Examined: all entries in the file which the user has displayed.

Examples:

RESTRICT FROM "COMSEVENTHFLT"

The above will select just those messages which are from COMSEVENTHFLT.

RESTRICT FROM "COMSEVENTHFLT" AND SUBJECT "TRAINING EVENT"

The above will select just those messages which are from COMSEVENTHFLT and at the same time have the string "TRAINING EVENT" in the Subject field.

RESTRICT FROM "J301" OR FROM "COMNAVMARIANAS GUAM"

The above will select only those messages which are from J301 or from COMNAVMARIANAS GUAM. These two groups of messages will then be displayed.

<STRING>

Allowed values: A string may be either a single word or a group of words, connected by punctuation or spaces, which MUST be enclosed by quotation marks.

Examples: "TRAINING", "TRAINING EVENT"

<USER ID>

Allowed values: Any user name or office code known to the SIGMA message service.

Example: SMITH or J301.

XII. INSTRUCTION DESCRIPTIONS

The following lists all instructions and the parameters needed in alphabetical order. Instructions which are shown hyphenated and surrounded by exclamation points are Function keys. All others must be typed in and must be completed with the EXECUTE key. Words in parentheses are optional. They may be entered or left out at the user's discretion. If they are left out, the message service will automatically fill them in. This occurs during the process of executing the instruction. Parameters in angle brackets must be replaced with an acceptable value. See the above material for acceptable values for each of the parameters used below.

ABORT

Abort operates on the Open object.

The effect of the Abort instruction is to eliminate any changes made to displayed text, a message or a file. Thus changes to these objects are not permanent if an Abort instruction is given.

Note how the instructions ABORT, FINISH and SAVE compare:

ABORT eliminates any changes made to an object during editing, closes the object and clears the screen. It works for files, messages, or text objects. The Abort instruction should be used, for example, when you realize you have asked for the wrong type of message form. It must be typed in.

FINISH retains all changes made, closes the object and clears the screen. It works for files, messages, or text objects. The Finish instruction key would be used when you have completed working with an object. This is also done automatically by SIGMA when you ask for another object of the same type. But if you are only going to work on one object, a file for instance, you would use Finish when you are through with it.

SAVE retains changes made to a text object, but does not close the text object, leaving it on the screen for further work. This instruction key would be used if you are going to leave the terminal for a moment, and you want to be sure your work on a text object up to that point is preserved.

Instruction Form: ABORT

Any changes or additions to the text, file, or message displayed are erased back to the point of the last SAVE or FINISH. The display window is also cleared.

ACTION

The user may assign or transfer action on any Transmitted message using the ACTION instruction. This causes a for_Action entry for the message to be delivered to the

Pending file of the specified user, and adds his name to the Action (or Cog or Orig) field of the message. It also automatically causes an entry in the ACTION_LOG reflecting this action assignment. The first form assigns action on the open message to a specific user. The second assigns action on the specified message to a specific user.

Instruction Forms: ACTION (MESSAGE) <USER ID>

ACTION (MESSAGE) <Message ID> <USER ID>

The next form forwards for action the next or current message in the open file. In the case where Next is used, the next entry becomes the current entry.

Instruction Forms: ACTION (ENTRY) NEXT <USER ID>

ACTION (ENTRY) CURRENT <USER ID>

The next form forwards for action a message that is identified by its entry number. This entry also becomes the current entry.

Instruction Form: ACTION (ENTRY) <NUMBER> <USER ID>

The next form forwards for action the message pointed to by the marked entry in the displayed file. The marked entry also becomes the current entry.

Instruction Form: ACTION (ENTRY) <HERE> <USER ID>

ALERT ON/OFF

When the key is pushed, SIGMA places an "Alert List" in the View window. If the key is pushed again the Alert List is removed and the previous state of the screen is returned. If there had been a View window, its contents are restored. If there was not a View window, the original displayed object is restored to full screen. If a VIEW instruction is executed while viewing the Alert List, the Alert List is replaced by the Viewed object. If the !!CLEAR_VIEW!! key is pushed while the Alert List is being viewed, the screen returns to full screen display window. The Alert List is not destroyed by !!CLEAR_VIEW!! or !!ALERT ON/OFF!! and may be displayed any time it is not on screen by pushing !!ALERT ON/OFF!! The Alert List is cleared by the RESET (Alerts) instruction.

AUGMENT

The user may add an additional subset of file entries to those presently displayed by using the Augment Selection instruction shown below.

Instruction Form: AUGMENT (SELECTION) <SELECTOR>

Example: AUGMENT SELECTION FROM "CINCPAC"

This instruction adds file entries to the currently displayed list of entries from the file open for display.

The user may choose to add any subset of file entries, according to the <SELECTOR> specification. If a given file entry is identified by more than one Display or Augment instruction, it will appear only once in the displayed list. Through the use of AUGMENT and RESTRICT, the user may isolate just those entries satisfying his needs.

The following instruction augments the current file entry display to include all entries specified by the given selection criteria.

Instruction Form: AUGMENT (WITH) <EXISTING SELECTOR NAME>

BACKUP

The user may establish a selection of file entries which occurred previously as the result of issuing a series of AUGMENT and RESTRICT instructions. The Backup Instruction key displays the previous set of file entries. Repeated issuance of the Backup instruction will regress to earlier such displays.

Instruction Key: !!BACKUP-ONE!!

The Backup instruction is useful to recover a previous display if the user has, for example, inadvertently asked for the addition or removal of too large a subset of file entries.

The following typed form of the Backup instruction restores the display to the original display before any RESTRICT or AUGMENT instruction.

Instruction Form: BACKUP ALL (SELECTION)

CHOP

The Chop Instruction keys are used to sign off the current message being reviewed, stating either approval or disapproval. The user must indicate either approval or disapproval by pressing either the !!CHOP-YES!! or the !!CHOP-NO!! key.

Instruction keys: !!CHOP-YES!!

!!CHOP-NO!!

CLEAR

The Clear-view Instruction key clears the view window, leaving the entire work area for editing. See VIEW on placing information in the view window.

Instruction Key: !!CLEAR-VIEW!!

COMMENT

The user may attach a comment to a file entry or a message field with the use of HERE and the Comment instruction. The user may first mark a field or entry with the HERE key and then issue the Comment instruction, or vice versa. Once both operations have been performed, the user presses the EXECUTE key. A blank line will appear in the display, below the field selected with the HERE marker. The user's name is displayed followed by the <ACCESS SPECIFICATION> of the comment. The access specification for a comment is either Public, Private, or one other logon name who may see the comment. The default access specification is Private unless otherwise specified. Once the comment has been attached to the field or entry, the user may enter any number of lines of text.

If the user deletes the entire contents of a comment, the comment will be deleted when the object is FINISHed.

Note that when a comment is placed on a file entry, it belongs just to that file. However, if the entry is FILEd or MOVEd into another file, the comment goes along with it.

Instruction Forms: COMMENT (ENTRY) <HERE> <ACCESS SPECIFICATION>

COMMENT MESSAGE <HERE> <ACCESS SPECIFICATION>

The user may also attach a comment to a file entry using the entry number adjacent to the entry he desires to comment. If a comment is already attached to this entry, then COMMENT: appears above the previous comment.

Instruction Form: COMMENT (ENTRY) <NUMBER> <ACCESS SPECIFICATION>

Example: COMMENT ENTRY 5 PUBLIC

The above instruction is issued in the presence of a display of entries.

The following forms Comment the next or current entry.

Instruction Forms: COMMENT (ENTRY) NEXT <ACCESS SPECIFICATION>

COMMENT (ENTRY) CURRENT <ACCESS SPECIFICATION>

COORDINATE

The Coordinate instruction sends the displayed message out for review simultaneously to those users whose names on the Chop list or the Release field have been marked with a HERE.

Instruction Key: !!COORDINATE-MESSAGE!!

The displayed message must be a Memo or AUTODIN message in preparation.

COPY

The Copy instruction forms described below allow the user to make a copy of text or an in-preparation message for purposes of displaying, editing, or filing it. (Also see PICKUP and MOVE.)

The !!COPY-TEXT!! Instruction key copies the marked text from the display or view window and puts it wherever the cursor is when the key is pressed. Unlike the !!PICKUP-TEXT!! and !!MOVE-TEXT!! keys, the !!COPY-TEXT!! key does not delete anything from the screen. The text is retained for a corresponding PUT instruction. If you use this instruction or a combination of !!COPY-TEXT!!, !!PICKUP-TEXT!!, or !!MOVE-TEXT!! twice in succession, the first text retained will be overwritten by the second one.

Instruction Key: <HERE> <HERE> <Cursor Location> !!COPY-TEXT!!

The following Instruction form makes a copy of a current open in-preparation message, places a File_Copy entry in the Pending file, and displays the new message in the Display window. The FROM field is changed, and a new DTG is assigned, but otherwise the new message is simply a copy of the original. The new message can then be edited as necessary. The open object must be a message in preparation.

Instruction Form: COPY (MESSAGE)

The following typed instruction form makes a copy of the identified message in preparation, places a File_Copy entry in the Pending file, and displays the new message in the Display window. This instruction requires that no message be open for editing.

Instruction Form: COPY (MESSAGE) <Message ID>

The following keys create a copy of the next, the current, or the marked message in an open file. It requires that a file be displayed and that the indicated message be a message in preparation. The newly created copy is displayed in the Display window, and a File_Copy entry is placed in the Pending file.

Instruction Key: !!COPY-NEXT!!

!!COPY-ENTRY!!

<HERE> !!COPY-ENTRY!!

Instruction Forms: COPY (ENTRY) <NUMBER>

The following typed instruction forms copy the marked text from any window into the named text object. In reference to the first form shown below, any previously existing text in the named text object is destroyed.

Instruction Forms: COPY (TEXT) <HERE> <HERE> <EXISTING TEXT NAME>

COPY (TEXT) <HERE> <HERE> <NEW TEXT NAME>

CREATE

The Create instruction is used to create a new file, message, text object or Selector. These instructions are explained below.

The user may create a new file using the instruction form shown below. Files are created only in this explicit way. That is, a file must be created before messages are placed into it. The result of entering a file creation instruction is to establish a new file containing no entries. The newly created file is not displayed. For example, if you are looking at the Pending file and want to file entries from it into some new file, you can CREATE the new file without losing your display of the Pending file. You can then FILE entries from the Pending file into the new file.

Instruction Form: CREATE FILE <NEW FILE NAME> <SECURITY>

Example: CREATE FILE SHIPS UNCLASSIFIED

The user may create a new message and display the blank message form (for purposes of entering the message information) by using the following instruction. If there is already an open message, an automatic FINISH is done on it, whether or not the message is actually displayed.

Instruction Form:

CREATE MESSAGE <MESSAGE TYPE> <SECURITY> (<EXISTING FILE NAME>)

The optional <Existing File Name> allows you to name a file into which the new message is filed as it is created. If this parameter is omitted, SIGMA will default the file name to Pending. That is, if you type the instruction CREATE SECRET AUTODIN MESSAGE the new message will be filed into your Pending file automatically.

Messages which are not filed in any file are only recallable by drafter's name and DTG, which you must have written down.

The following instruction is used to create a text object and open it in the display window for editing. Again, the security default level is Unclassified.

Instruction Form: CREATE TEXT <NEW TEXT NAME> <SECURITY>

The following Create instruction saves the selector which describes the current display of the open file.

Instruction Form: CREATE SELECTOR <NEW SELECTOR NAME>

The following Create instruction creates a selector with the name given as defined by the given selector parameter.

Instruction Form:

CREATE SELECTOR <NEW SELECTOR NAME><SECURITY> <SELECTOR>

CURRENT-ENTRY

The !!CURRENT-ENTRY!! instruction key is used to set a new Current entry. Whichever entry is marked with a <HERE> will become the new Current one. If no <HERE> is supplied, this key just causes the screen to scroll until the current entry is on-screen, then puts the cursor on the current entry.

Instruction key: <HERE> !!CURRENT-ENTRY!!

DELETE

The family of Delete instructions permits the user to delete a file, file entries, a text object, or a selector. (You cannot delete a message, although you can effectively discard it if it is still in preparation. This is done with an ABORT. See FILE SYSTEM section.) These instructions are explained below.

The user may delete an existing file using the instruction form shown below. The result of deleting a file is to destroy the file and any entries it may contain. However, if this instruction is entered by accident, the file may be restored by using the RESTORE instruction. A deleted file is shown in the File Directory with an asterisk to the left of its name.

Instruction Form: DELETE FILE <EXISTING FILE NAME>

Example: DELETE FILE SHIPS

If the file being deleted is displayed at the time the Delete File instruction is issued, then the screen is erased. Otherwise, the screen remains unchanged.

The instruction below deletes a text object. If this instruction is entered by accident, the text object may be restored by using the RESTORE instruction. A deleted text object is shown in the Text Directory with an asterisk at the left of its name.

Instruction Form: DELETE TEXT <EXISTING TEXT NAME>

The following instructions delete file entries. The first one below deletes a file entry pointed to in the open file. If the instruction is entered by mistake, entries may be restored by using the RESTORE instruction. However, once a FINISH instruction is given, an entry is permanently deleted from the file.

Instruction key: <HERE> !!DELETE-ENTRY!!

The following Instruction keys each delete an entry in the open file.

Instruction keys: !!DELETE-NEXT!!

!!DELETE-ENTRY!!

The next typed instruction deletes a list of entries from the open file.

Instruction Form: DELETE (ENTRY) <ENTRY LIST>

The next instruction form deletes highlights from the text of an incoming AUTODIN Message (highlights cannot appear in other message types--see the HIGHLIGHT instruction). You can only delete those you create or those personally directed to you.

Instruction Form: DELETE HIGHLIGHT <HERE>

The next instruction forms delete highlights from a file entry. You can only delete those you create or those personally directed to you.

The following form deletes the highlights of the marked entry.

Instruction Form: DELETE HIGHLIGHT <HERE>

The following form deletes the highlights of the designated entry.

Instruction Form: DELETE HIGHLIGHT (ENTRY) <ENTRY NUMBER>

The following form deletes the highlights of the current entry.

Instruction Form: DELETE HIGHLIGHT (ENTRY) CURRENT

The following form deletes the highlights of the next entry.

Instruction Form: DELETE HIGHLIGHT (ENTRY) NEXT

The final Delete instruction, below, deletes a selector. If this instruction is entered by mistake, the selector may be restored by using the RESTORE instruction.

Instruction Form: DELETE SELECTOR <EXISTING SELECTOR NAME>

DIRECTORY

Each user has three directories which are lists of his files, defined selectors, and text objects. Each of these directories can be shown in the View window for reference only. The following Instruction keys show the corresponding directory in the View window.

Instruction keys:!!FILE-DIRECTORY!!

!!SELECTOR-DIRECTORY!!

!!TEXT-DIRECTORY!!

DISPLAY

The Display instructions place information in the display window. The various instructions below allow the user to display (for editing) files, messages, and text objects. If an object of the same kind is already open, an automatic FINISH is done and the object is closed. The first two forms shown below apply to files.

The user may display file entries (and their associated comments, if any) using the following instruction:

Instruction Form: DISPLAY FILE <EXISTING FILE NAME> (<SECURITY>)

Example: DISPLAY FILE SHIPS

The Display instruction, as opposed to the VIEW instruction, places the file contents in the display window where the user may further work with it. For example, the user may add or delete comments associated with the entries. The <SECURITY> parameter is

optional. Normally, it is not included and the file is opened at the level at which it was defined (see the CREATE instruction). If the file is a Pending file, the <SECURITY> parameter can be used to open the file at the security levels up to and including the given level. If the <SECURITY> parameter is NOT given for the Pending file, it is opened up to and including the SECRET level.

The following instruction key brings the display of the open file back into the display window.

Instruction key: !!SHOW-FILE!!

The next form displays the special date files (described in FILE SYSTEM) which contain all messages with DTG's of a given date:

Instruction Form: DISPLAY (DATEFILE) <DATE> <SECURITY>

The following two forms display text objects. The first displays a specified text object and the second redisplays the current open text object.

Instruction Forms: DISPLAY TEXT <EXISTING TEXT NAME>

!!SHOW-TEXT!!

The following forms display messages. The first displays the open message.

Instruction key: !!SHOW-MESSAGE!!

The following typed instruction form displays the message identified by the <Message ID>. When the user types the instruction DISPLAY MESSAGE without supplying the <Message ID> parameter, SIGMA will display the last message the user displayed during this session (if there was one). That is, the instruction DISPLAY MESSAGE defaults to the last displayed message, even if it has been FINISHed, or otherwise closed.

Instruction Form: DISPLAY (MESSAGE) <Message ID>

The following instruction keys display the next or current message in the open file.

Instruction keys: !!DISPLAY-NEXT!!

!!DISPLAY-ENTRY!!

The following instruction key displays the message pointed to in the open file.

Instruction key: <HERE> !!DISPLAY-ENTRY!!

The next typed form displays the numbered entry in the open file.

Instruction Form: DISPLAY (ENTRY) <NUMBER>

Any of the above forms which Display a message (including all the Display Entry forms) may ask for further acknowledgment if the message requested has been archived. In this case, the user is asked to acknowledge (with YES, NO or CANCEL) that the message should be "retrieved" from the archive. Retrieval may take some time, so SIGMA allows the user to proceed with other things. When the requested message has been retrieved, SIGMA notifies the user with a Retrieved entry in his Pending file.

EMPTY

The entries in a user's file may all be deleted in one operation without having to DISPLAY the file using the EMPTY instruction. If the file is open at the time, EMPTY causes it to be FINISHed. Once a file has been EMPTYd its entries cannot be RESTOREd. Only the person who created the file may EMPTY it.

Instruction Form: EMPTY (FILE) <EXISTING FILE NAME>

EXERCISE

After instructing the message service to begin a LESSON on the use of the service, a user may do a number of exercises dealing with the instructions of the particular lesson.

Within the exercise, the instruction QUIT is available to leave the exercise and the lesson, and return to the normal SIGMA service. In each case, detailed instructions are given on how to perform and leave the current exercise.

Instruction Form: EXERCISE (NUMBER) <NUMBER>

FILE

The following five instruction forms are used to file messages and file entries. The first three forms shown below require that the displayed object be a file. The fourth requires that the displayed object be a message.

The first form below files the marked entry from the displayed file into another file.

Instruction Form: FILE (ENTRY) <HERE> <EXISTING FILE NAME>

The next instruction form files a list of entries from the open file into another file.

An entry list as the name implies is a list of one or more entries, separated by commas, such as 2,4,9. A sequence may be given separated by dashes, such as 6-12 which signifies entries 6 through 12. The comma and dash may be used in the same entry list, such as 2,4-7,10.

Instruction Form: FILE (ENTRY) <ENTRY LIST> <EXISTING FILE NAME>

The next instruction form, like the preceding ones, files a list of entries from the open file into the named file; but here entries in the named file are arranged so those with the older DTG's precede more recent ones, i.e., the entries of the named file are sorted by DTG. The named file is inaccessible to you until the sort is completed. If you do not own the named file the entries will be filed but not sorted ; in that case you will be informed the sort was not possible.

Instruction Form: FILE (ENTRY) <ENTRY LIST> SORT <EXISTING FILE NAME>

The following forms file the next or current entry.

Instruction Forms: FILE (ENTRY) NEXT <EXISTING FILE NAME>

FILE (ENTRY) CURRENT <EXISTING FILE NAME>

The last instruction forms file the message being edited into a file, either the file identified in the instruction or the open file.

Instruction Forms: FILE (MESSAGE) <EXISTING FILE NAME>

FILE (MESSAGE)

FIND

The Find instruction locates the next occurrence of a specified string of text following the HERE marker. This instruction works on any displayed object. When used with a file, FIND <STRING> does not affect the designation of the entry marked CURRENT. Also, FIND <STRING> may not work correctly in a file if the string being found is too far away from what is being displayed on the screen. This is due to an implementation detail.

A string may be either a single word or may be a group of words, connected by punctuation or spaces, which MUST be enclosed by quotation marks. Examples are: "FLASH" and "FLASH OVERRIDE." Upper and lower case are identical to the service.

Instruction Form: FIND (STRING) <STRING> <HERE>

The <HERE> parameter is only needed for repeated FIND STRING instructions. That is, the first time a particular FIND STRING is performed, SIGMA will find the first occurrence

from the top of the displayed object. If you then want to find the next occurrence, you must place a <HERE> mark at or below the first place found in the object. Each subsequent FIND STRING instruction will find the next occurrence below the <HERE> mark.

The next instruction finds the first occurrence of the specified string, beginning with the initial part of the text.

Instruction Form: FIND (STRING) <STRING>

The following instruction finds and displays the file entry with the number requested. The newly displayed entry now becomes the current entry.

Instruction Form: FIND (ENTRY) <NUMBER>

The following two typed instruction forms cause the displayed object to be scrolled until the top (or bottom) of the object appears on screen. If the screen has both a DISPLAY and VIEW window showing, putting a HERE marker into the VIEW window prior to executing will cause the View window to scroll.

Instruction Forms: FIND TOP

FIND BOTTOM

FINISH

This Instruction key terminates editing of the current object (file, message, or text), saves the result, and closes the object (thus clearing if off the screen).

Instruction key: !!FINISH!!

Note how the instructions ABORT, FINISH and SAVE compare:

ABORT eliminates any changes made to an object during editing, closes the object and clears the screen. It works for files, messages, or text objects. The Abort instruction should be used, for example, when you realize you have asked for the wrong type of message form. It must be typed in.

FINISH retains all changes made, closes the object and clears the screen. It works for files, messages, or text objects. The Finish Instruction key would be used when you have completed working with an object. This is also done automatically by SIGMA when you ask for another object of the same type. But if you are only going to work on one object, a file for instance, you would use Finish when you are through with it.

SAVE retains changes made to a text object, but does not close the text object, leaving it on the screen for further work. This Instruction key would be used if you are going to leave the terminal for a moment, and you want to be sure your work on a text object up to that point is preserved.

FORWARD

The Forward instructions permit the user to forward messages to other users. A for_info entry is placed in the Pending file of the user to whom the message is being forwarded, and his name (or title) is added to the Internal Distribution field shown with the message.

The first Forward form, shown below, sends the open message to the specified user.

Instruction Form: FORWARD (MESSAGE) <USER ID>

The next Forward form sends the specified message to the identified user.

Instruction Form: FORWARD (MESSAGE) <Message ID> <USER ID>

The next forms forward the next or current message in the open file. It requires that a file be displayed.

Instruction Forms: FORWARD (ENTRY) NEXT <USER ID>

FORWARD (ENTRY) CURRENT <USER ID>

The next form forwards a message that is identified by its entry number.

Instruction Form: FORWARD (ENTRY) <NUMBER> <USER ID>

The next form forwards the message marked in the displayed file.

Instruction Form: FORWARD (ENTRY) <HERE> <USER ID>

GET

The Get instructions allow the user to access information belonging to other users in their access control group. Through the use of these instructions, one can effectively access readboards.

The instruction below gets another user's file and names it for your use. The user-id is the identifier of a user other than yourself. It may be either an office code or a personal name. You will receive an error message if you attempt to GET a file to which you are not permitted access.

In the case of GET FILE, the file you get will always reflect the current state of the foreign file. That is, whenever you look at it in the future, you will see any changes the owner of the file has made to it, even if they were made after you did the GET.

Instruction Form:

GET FILE <EXISTING FILE NAME> (FROM) <USER ID> <NEW FILE NAME>

Example: GET FILE BLUEEAGLE FROM J3 BLUE_EAGLE_TRAINING_FILE

The next instruction gets another user's file for your use and calls it by the same name.

Instruction Form: GET FILE <EXISTING FILE NAME> (FROM) <USER ID>

In the cases of GET SELECTOR and GET TEXT, the object you get is your own personal copy of the owner's object as of the moment you do the GET. It will not reflect any changes the owner makes after you have done the GET.

The next instruction gets a user's selector and names it for your use.

Instruction Form:

GET SELECTOR <EXISTING SELECTOR NAME> (FROM) <USER ID> <NEW SELECTOR NAME>

The next instruction gets another user's selector for your use and retains it by the same name.

Instruction Form:

GET SELECTOR <EXISTING SELECTOR NAME> (FROM) <USER ID>

Similarly, the next two instructions obtain another user's text objects.

Instruction Forms:

GET TEXT <EXISTING TEXT NAME> (FROM) <USER ID> <NEW TEXT NAME>

GET TEXT <EXISTING TEXT NAME> (FROM) <USER ID>

GO-TO-NEXT

The !!GO-TO-NEXT!! Instruction key is used to make the Next entry the new Current entry.

Instruction key: !!GO-TO-NEXT!!

HELP

The HELP Instruction key is normally used to access the on-line form of this reference information. When it is pressed, the Instruction Line, Display window and View window (if any) are cleared, and a special display is shown. Just below the Status Line is the message: "HELP: Hit CANCEL key to get back to normal screen". This tells you that you are looking at HELP information, and that you can return to the normal Instruction Line, Display window and View window as they were before you pressed the HELP key by pressing the CANCEL key.

The next line includes two hilighted terms "HELP" and "SERVICE FACILITIES" (shown as black on white instead of white on black), and a field called "Current Term". On-line HELP is accessed on the basis of "terms". Each term is some word, phrase, Instruction or concept which describes something about SIGMA. The word or phrase shown as the Current Term is the one described by the HELP information shown. The two special terms "HELP" and "SERVICE FACILITIES" can be accessed by moving the cursor into the appropriate highlighted area on this line and pressing the EXECUTE or HELP key. The term "HELP" describes how to use the HELP facility. The term "SERVICE FACILITIES" gives a list of SIGMA facilities which can be used as an index to find further information. In general, words or phrases shown in highlighted form in the HELP text can be selected as other terms to be described. The user simply moves the cursor into any such highlighted term and presses EXECUTE or HELP. For example, if the user asks to see the term SERVICE FACILITIES, it shows a number of other highlighted words and phrases in index form. The user can select any of these other terms by moving the cursor into the desired highlighted area and pressing EXECUTE or HELP.

The last line above the actual HELP text shows two highlighted areas "BACK", and "FORWARD", and a highlighted field called "New Term". If the user wants help with some term not shown on the screen, he can type the term at New Term and press EXECUTE or HELP. Once the user has asked for help with more than one term, he can go back to see the help text for previous terms by moving the cursor into the highlighted word BACK and pressing EXECUTE or HELP. If he has already gone back, he can then go forward again by selecting the word FORWARD. The HELP text is saved for a maximum of three previous terms.

The HELP key is used for a different purpose when the user is taking a Lesson. The normal Help facility is unfortunately not available during a Lesson. The key is used instead to switch back and forth between the text of an Exercise and the normal screen that results from trying one of the Instructions in the Exercise.

HIGHLIGHT

The user may highlight text in a Transmitted message with the use of HERE and the HIGHLIGHT instruction. The user first marks the first and last characters to be highlighted with the HERE key and then issues the highlight instruction. Once both operations have been performed, the user presses the EXECUTE key. This causes the text indicated to be presented in inverse video. The user may specify access control for the highlight to be Public, Private, or one other logon name who may see the highlight. The default access specification, which is used if no other access is specified, is Private. A highlight may be deleted by indicating it with a HERE somewhere within the highlighted text and issuing a

DELETE HIGHLIGHT instruction. Only the person who creates a highlight may delete that highlight. If a personal highlight is generated for a second user, that user may delete the highlight from his own view; i.e., the highlight is converted to Private for the user who created it.

Instruction Form: HIGHLIGHT (MESSAGE) <HERE> <HERE> <ACCESS SPECIFICATION>

The user may also attach a highlight to a file entry using HERE or the entry number to specify the entry he desires to highlight. A highlighted entry is shown with the Subject line of the entry in inverted video. A user may delete a highlight from an entry by executing a DELETE HIGHLIGHT instruction with a HERE or the entry number.

Note that when a highlight is placed on a file entry, it belongs just to that file. However if the entry is FILEd or MOVEd into another file, the highlight goes along with it.

Instruction Form: HIGHLIGHT (ENTRY) <HERE> <ACCESS SPECIFICATION>

HIGHLIGHT (ENTRY) <NUMBER> <ACCESS SPECIFICATION>

Example: HIGHLIGHT ENTRY 5 PUBLIC

The following forms highlight the next or current entry.

Instruction Forms: HIGHLIGHT (ENTRY) NEXT <ACCESS SPECIFICATION>

HIGHLIGHT (ENTRY) CURRENT <ACCESS SPECIFICATION>

IDENTIFY

The Identify instruction is used to change the previous user's identity to that of another. (See SERVICE FEATURES.)

Instruction Form: IDENTIFY (AS) <USER ID>

KEYWORD

The Keyword instructions below allow the user to tag file entries with keywords or a keyword string defined by the user. At present keywords are not shown, so they must be written down or remembered. These keywords may later be used as selectors. (See RESTRICT instruction.) Note that if a file has already been restricted when the Keyword instruction is performed, only those entries which are displayed (that is, those in the restricted subset) will be affected by the Keyword.

Of the following forms of the instruction, only the forms which use HERE and NEXT result in changing the Current entry in the file.

Instruction Forms: KEYWORD (ENTRY) <HERE> <STRING>

KEYWORD (ENTRY) <ENTRY LIST> <STRING>

KEYWORD (ENTRY) CURRENT <STRING>

KEYWORD (ENTRY) NEXT <STRING>

Example: KEYWORD ENTRY NEXT "SHIPPING"

The above entry could then be found with the instruction:

RESTRICT KEYWORD "SHIPPING"

LESSON

SIGMA provides a number of lessons which the user can take to learn to use the service. The lessons consist of descriptive material, plus the option to do an EXERCISE using the instructions discussed in the lesson.

Note that you must perform a FINISH or an ABORT on any open file, message and text object before you can take a lesson, and you must also press the Clear-View key if there is anything shown in the View window.

Lessons have numbers and parts, such as Lesson 2 Part B. Available lessons are:

Lesson 1: A General Description of the SIGMA Service

Lesson 2 A: Beginning to Use the SIGMA Service

Lesson 2 B: Beginning to Use the SIGMA Service

Lesson 3 A: The Filing Service - Basic Filing Techniques

Lesson 3 B: Advanced Filing Techniques

Lesson 3 C: Special Files

Lesson 4: Message Reception and Distribution

Lesson 5 A: Text Objects

Lesson 5 B: Editing Instructions for Text Objects

Lesson 5 C: Editing Instruction Keys

Lesson 6: Message Drafting

Lesson 7: Message Review and Coordination

The following instruction is used to take a lesson.

Instruction Form: LESSON <NUMBER> <LETTER>

Note that if the letter part of the Lesson is left off, part "A" is assumed.

You may leave a lesson at any time and return to the normal SIGMA service using the QUIT instruction. (When you QUIT a lesson, any objects opened during an exercise are automatically ABORTed.)

LOG

SIGMA is started up by turning on or resetting the terminal. The user then fills in the required information on the screen in order to log on to the message service. The information which must be filled in includes your log on name (which may be an office code), your log on password, and your requested maximum security level. If you are logging on as an office code, you must also fill in your personal identity (name) and personal password. To fill in these items on the screen, use the cursor movement, editing keys, and then press the EXECUTE key.

If you do not fill in the required information within a few minutes, the login form will automatically disappear. In order to get it back again, the terminal must be Reset.

The Log Off instruction logs the user off of the message service.

Instruction Form: LOG (OFF)

When you LOG OFF, all open objects are automatically FINISHed for you.

MOVE

The MOVE instructions allow moving file entries or text.

The Move Entry instructions move entries from file to file. In each case the displayed object must be a file. The first form shown below moves the marked entry out of the open file (deleting it from the open file) and into another, named file.

Instruction Form: MOVE (ENTRY) <HERE> <EXISTING FILE NAME>

The next form files a list of entries in the named file and deletes them from the open file. Use the FILE instruction if the entries are not to be deleted.

Instruction Form: MOVE (ENTRY) <ENTRY LIST> <EXISTING FILE NAME>

The next instruction form, like the preceding ones, files a list of entries into the named file and deletes them from the open file; but here entries in the named file are arranged so those with the older DTG's precede more recent ones, i.e., the entries of the named file are sorted by DTG. The named file is inaccessible to you until the sort is completed. If you do not own the named file the entries will be filed but not sorted; in that case you will be informed the sort was not possible.

Instruction Form: MOVE (ENTRY) <ENTRY LIST> SORT <EXISTING FILE NAME>

The following forms move the next or current entry.

Instruction Forms: MOVE (ENTRY) NEXT <EXISTING FILE NAME>

MOVE (ENTRY) CURRENT <EXISTING FILE NAME>

The !!MOVE-TEXT!! instruction key allows the user to move text from one place on the screen to another in one operation. It picks up the text between the two places marked with the <HERE> key, deletes it from the edit window, and places it at the cursor position. The picked up text is saved and can be placed elsewhere with the !!PUT-TEXT!! instruction key. Doing a !!PUT-TEXT!! does not destroy the saved text, so that you can put the same text in several different places. Note that whenever you do another !!MOVE-TEXT!!, !!PICKUP-TEXT!!, or !!COPY-TEXT!! the saved text will be overwritten by the new text picked up.

Note how this instruction compares with the COPY instruction. COPY only makes a copy of the marked object. MOVE not only makes a copy of the marked object, but also deletes it from the screen. However, if the text marked to be moved is in the View window or for some other reason is NOT text which can be edited, the MOVE-TEXT key will perform the equivalent of a COPY-TEXT only. That is, it will NOT delete it from the View window.

Instruction key: <HERE> <HERE> <CURSOR LOCATION>!!MOVE-TEXT!!

PICKUP

The Pickup instructions allow picking up text between two HERE marks on the screen and saving the text to be Put somewhere else. The Picked up text is deleted from the screen.

The following Instruction key Picks up text between two places on the screen marked with the <HERE> key. The text picked up in this way can be Put somewhere else

later on (inserted elsewhere on the screen) with the PUT Instruction key. The text Picked up is available to be Put elsewhere until the next time a !!PICKUP-TEXT!!, !!COPY-TEXT!!, or !!MOVE-TEXT!! is done.

Instruction key: <HERE> <HERE> !!PICKUP-TEXT!!

The next form is typed in. It deletes the marked text from the display window and saves it in a previously named text object, replacing the existing text with the new text.

Instruction Form: PICKUP (TEXT) <HERE> <HERE> <EXISTING TEXT NAME>

The form shown below picks up the marked text from the display window and saves it in a new text object. It has the effect of creating a text object with the new name, without the need to CREATE TEXT beforehand.

Instruction Form: MOVE (TEXT) <HERE> <HERE> <NEW TEXT NAME>

The security level of the new text object will be the same as the one containing the text to be moved.

PRINT

The Print instructions make printed copies of information on the line printer. The various instructions below allow the user to print messages and text objects. Of the following Instruction forms, only PRINT (ENTRY) <NUMBER>, <HERE> !!PRINT-ENTRY!!, and !!PRINT-NEXT!! result in changing the Current entry in the file.

The following forms print a specified text object.

Instruction Forms: PRINT (TEXT) <EXISTING TEXT NAME>

To print your own text directory, you first use the TEXT-DIRECTORY Instruction key, and then use the Instruction key:

!!PRINT-VIEW!!

The following form prints some other user's text directory and the two forms following print the file directory and the selector directory, respectively.

Instruction Forms: PRINT TEXT DIRECTORY <USER ID>

PRINT FILE DIRECTORY <USER ID>

PRINT SELECTOR DIRECTORY <USER ID>

The following forms print messages. The first prints the open message.

Instruction Form: PRINT MESSAGE

The following instruction form prints the message identified by the Message ID.

Instruction Form: PRINT (MESSAGE) <Message ID>

The following instruction keys print the next or current message in the open file.

Instruction keys: !!PRINT-NEXT!!

!!PRINT-ENTRY!!

Similar to the above, the following Instruction key prints the message selected in the open file and makes it the Current entry.

Instruction key: <HERE> !!PRINT-ENTRY!!

The next form prints the message associated with the numbered entry in the open file. It requires that the displayed object be a file.

Instruction Form: PRINT (ENTRY) <NUMBER>

The following instruction prints the open file itself, as it appears on the screen.

Instruction Form: PRINT OPEN FILE

The following instruction prints the named selector.

Instruction Form: PRINT SELECTOR <EXISTING SELECTOR NAME>

The next two Instruction keys print the displayed object and the viewed object, respectively. In the case of Print-View, if the viewed object is a file, the printed output *may be less than the full file contents.*

Instruction keys: !!PRINT-DISPLAY!!

!!PRINT-VIEW!!

The instruction form below prints a particular reviewer's version of a displayed message in preparation.

Instruction Form: PRINT (VERSION) <USER ID>

PROMPT

The PROMPT key can be used whenever SIGMA is unable to understand a typed instruction. It shows all alternate forms of the instruction you have typed. If you have typed something which might be one of several different instructions, it shows the alternate forms of each legal instruction in the given context. It usually provides enough information for you to figure out why SIGMA is having trouble understanding what you typed. For example, when SIGMA tells you your instruction is incomplete, pressing PROMPT will show you the form of the instruction you are typing, including all necessary parameters.

If you want further detail about one of the alternate instructions shown when you press Prompt, you can move the cursor onto the line describing that instruction and press PROMPT again. This gives you more information about that particular instruction.

When you are taking a Lesson, the PROMPT key has a different use. Since Prompt is unfortunately not available during a Lesson, the key is used instead to return to the text of a Lesson when you are taking an Exercise.

PUT

The Put instruction moves text to a marked position in the display window. The named text object may be one which was created with a CREATE TEXT instruction, or it may have been picked up with the PICKUP instruction.

The following Instruction key takes the text Picked up by the last !!PICKUP-TEXT!!, !!MOVE-TEXT!!, or !!COPY-TEXT!! and puts it at the place marked with the <HERE>.

Instruction Form: <HERE> !!PUT-TEXT!!

The form below is typed in. It places the text identified by the given name at the spot marked by the <HERE>.

PUT (TEXT) <HERE> <EXISTING TEXT NAME>

QUIT

From within a LESSON or an EXERCISE use this instruction to return to the normal SIGMA service.

Instruction Form: QUIT (LESSON)

READDRESSAL

The Readdress instruction creates a special form of In-preparation message, which when released causes LDMX to send an AUTODIN message incorporating the specified message to the addresses identified in the readdressal message. In SIGMA the Readdressal message may be treated like any other message in preparation; the user fills in the addressees, adds appropriate comments for users who will review the Readdressal message, then sends it out for Coordination. Eventually the message will be Released by an authorized Releaser. The text of the Readdressal message is fixed form and is inserted automatically by SIGMA, but it may be edited in case corrections must be made to the Date Time Group or spelling of the Originator.

The following typed instruction form creates a Readdressal message for the open message, places a File_Copy entry in the Pending file, and displays the message in the Display window. For this instruction form the open object must be a received AUTODIN message.

Instruction Form: READDRESS (MESSAGE)

The following typed instruction form creates a Readdressal message for the specified message, places a File_Copy entry in the Pending file, and displays the message in the Display window for editing.

Instruction Form: READDRESS (MESSAGE) <Message ID>

The following typed instruction creates a Readdressal message for the indicated message, places a File_Copy in the Pending file for the user message and displays the message. This instruction form requires that a file be displayed and that the indicated message be a received AUTODIN message.

Instruction Form: READDRESS (ENTRY) <NUMBER>

The following instruction keys create a Readdressal message for the indicated message (current, next, or the one marked with a !!HERE!!), places a File_Copy entry in the Pending file for the new message, and displays the message in the Display window for editing. This instruction form requires that a file be displayed and that the indicated message be a received AUTODIN message.

Instruction Key: !!READDRESS-ENTRY!!

!!READDRESS-NEXT!!

<HERE> !!READDRESS-ENTRY!!

RECLASSIFY

The Reclassify instruction changes the security level of a text object. It requires a security confirmation. You cannot change the security level of a message, file, or selector with SIGMA.

Instruction Form: RECLASSIFY (TEXT) <EXISTING TEXT NAME> <SECURITY>

RELEASE

The Release Instruction key causes the displayed message in preparation to be transmitted. If it is an AUTODIN message it is sent to LDMX. Memos and Notes are sent to the designated addressees. SIGMA checks local addressees (for Memos and Notes) and message precedence (for AUTODIN) when you press Release, and asks that you reconfirm by pressing the !!YES!!, !!NO!!, or !!CANCEL!! key. The message is transmitted to all addressees.

Instruction key: !!RELEASE-MESSAGE!!

REPLY

The Reply instructions permit the user to create a message in response to an earlier message. It differs from the CREATE instruction in that the To field of the created message is automatically filled in with the contents of the From field of the earlier message. Similarly, the Info field is copied. The References field of the created message will contain the Message ID of the earlier message. The default for the security level of the new message is the same as that of the earlier message it references. A File_Copy is placed in the Pending file.

Note that Reply simply creates a new message. The new message must then be completed and Released in order for it to be sent.

The first Reply form, shown below, creates a new message for editing and composition, in reply to the specified message.

Instruction Form:

REPLY (MESSAGE) <Message ID> <MESSAGE TYPE> <SECURITY>

The next forms are key Instructions. They create a reply to the next or current message in the open file. They require that an open file be displayed. They default the <MESSAGE TYPE> and <SECURITY> level of the new message to be the same as the one being replied to. Finally, a File_Copy entry is automatically placed in the Pending file.

Instruction key: !!REPLY-NEXT!!

!!REPLY-ENTRY!!

The next Instruction key creates a reply to the message marked in the displayed file.

Instruction key: <HERE> !!REPLY-ENTRY!!

The next form is typed in. It creates a reply to the file entry that is identified by its entry number.

Instruction Form:

REPLY (ENTRY) <NUMBER> <MESSAGE TYPE> <SECURITY>

RESET

The Reset (Alerts) instruction clears the contents of the Alert List and resets the Flash Line Alert count to zero.

RESTORE

The Restore instructions restore text objects, file entries, entire files, and selectors that have been deleted during the session (see DELETE). Should a new object with the same name as a deleted object be created during the session, the deleted object is rewritten, thus destroying it altogether and making it unrestorable.

The first form below is typed. It restores a deleted text object to its original state.

Instruction Form: RESTORE TEXT <EXISTING TEXT NAME>

The next form is typed. It restores a list of entries in the open file.

Instruction Form: RESTORE (ENTRY) <ENTRY LIST>

The next form is typed. It restores a deleted file to its original state.

Instruction Form: RESTORE FILE <EXISTING FILE NAME>

The last form, shown below, restores a specified selector which has been deleted.

Instruction Form: RESTORE SELECTOR <EXISTING SELECTOR NAME>

RESTRICT

The user may restrict the display to any subset of those file entries presently displayed by using the Restrict Selection instruction shown below.

Instruction Form: RESTRICT (SELECTION) <SELECTOR>

Example: RESTRICT FROM "PACAF"

The above instruction restricts the current displayed list of file entries to those originated by PACAF.

The following form restricts the current file display by the specified selector.

Instruction Form: RESTRICT (WITH) <EXISTING SELECTOR NAME>

Example: RESTRICT WITH Old_Selector

The user may choose to keep any subset of entries, described by the SELECTOR specification.

Through the use of the RESTRICT and the AUGMENT instructions, the user may isolate just those entries satisfying his needs.

ROUTE

This is a special-purpose instruction which allows you to specify routing for a number of messages at once. Routing is the distributing of messages (via Action and Forward instructions) to other SIGMA users and filing them into various files. The Route instruction provides a convenient way of doing this for standard groups of users and files.

Suppose your Pending file has a number of messages in it which are related to the Elegant Eagle exercise, and that for each of these messages you want to ACTION J3 and FORWARD the message to J321, J323 and J324. At the same time you might want to FILE all these messages into a special file called ELEGANT_EAGLE and also into a file called EXERCISE_FILE, and delete them from your Pending file. All these operations can be done with a single ROUTE instruction.

Instruction form: ROUTE <Entry List> <Existing Text Name>

The instruction assumes you have an open file from which you are going to select entries to be ROUTEd. You can DISPLAY this file (which might be your Pending file or any other file) and use the RESTRICT and AUGMENT instructions to select a subset of its entries, or you can refer directly to those entries you want by giving their numbers in an <Entry List>. For example, if you know that the messages related to the Elegant Eagle

exercise are entries 1,3,5,6,7,8, and 13, you might use the <Entry List> 1,3 5-8,13. If all these messages had the words "Elegant Eagle" in their Subject fields, you might instead use the instruction:

RESTRICT SUBJECT "Elegant Eagle"

which would restrict the display to only those entries. Similarly, if you had previously used the instruction KEYWORD <Entry> "Elegant Eagle" on each of these entries, you could now produce the same restricted subset with the instruction RESTRICT KEYWORD "Elegant Eagle".

Once you have decided on the <Entry List> to be ROUTEd, either by typing entry numbers explicitly or by Restricting or Augmenting and then using a hyphen for the <Entry List>, you need to specify a Text object. This must be an Unclassified text object whose contents is the series of operations to be performed by the ROUTE instruction.

To continue the above example, suppose you create a Text object named Elegant_Eagle_Group which has the following contents:

* Standard group for routing messages related to Elegant Eagle *

ACTION: J3.

FORWARD: J321,J323,J324. *But not to J322*

FILE: Elegant_Eagle,Exercise_File.

DELETE: Yes.

You can also specify the full ROUTE instruction for the above example in either of the following two ways:

ROUTE 1,3,5-8 13 Elegant_Eagle_Group

ROUTE - Elegant_Eagle_Group

where the latter form could be used if you had already selected the entries 1,3,5,6,7,8,13 to be the restricted display.

In either case, the ROUTE instruction will apply each of the operations specified in the Text object to each of the entries. The operations allowed within a Text object to be used with the ROUTE instruction are:

ACTION: Assign Action, usually to a single user.

FORWARD: Send info copies of the message to a list of users.

FILE: File copies of the entries into one or more files.

DELETE: This option allows the ROUTEd entries to be deleted.

Any or all of the above operations may appear in a given ROUTE Text object, and they may appear in any order.

NOTE:

Each operation MUST BE FOLLOWED BY A COLON, as shown above.

Each operation MUST BE TERMINATED BY A PERIOD, as shown above.

Comments can be written in the Text object by placing them between asterisks (these might be used as explanatory notes for a particular Routing). Two comments are shown in the example: one on the first line by itself, and one following the FORWARD operation.

The ACTION operation behaves like the Typed ACTION instruction: for each message in the <Entry List> it puts an entry in the Pending file of the user following the colon. It also puts an entry in the Action_Log and modifies the Action, Cog, or Orig field of the message as necessary. If the user performing the ROUTE specifies an ACTION operation and does not have an Action_Log file, SIGMA will not perform the ROUTE instruction.

The FORWARD operation behaves like the Typed FORWARD instruction, except that more than one user name can be specified at once. The message will be forwarded to all of the given users, and their names will appear on the message's distribution list.

The FILE operation behaves like the Typed FILE instruction, except that more than one file can be specified at once, separated by commas. The message will be filed into all of the given files.

Whether or not any other operations are specified in a ROUTE Text Object, the user may have a "DELETE: YES." operation which will delete all the entries in the <Entry List> from the open file. (The user need never specify the operation "DELETE: NO.", since entries will only be deleted if the DELETE operation has a "YES" in it.)

When first creating a Text object for use with ROUTE, you may want to Display the Text object while trying out the ROUTE instruction. If there are any errors in the Text object's ROUTE operations, you can then edit the Text object in the display window and retry the ROUTE operation. When a Text object has been used successfully once with a ROUTE instruction, you can use it again any time you want the same ACTION, FORWARD, FILE and DELETE operations to be performed on some new set of entries. It may still be helpful to Display the Text object before doing the ROUTE to make sure the operations it specifies are applicable to the new set of entries.

SAVE

The Save instruction key preserves present editing of text objects only.

Note how the instructions ABORT, FINISH and SAVE compare:

ABORT eliminates any changes made to an object during editing, closes the object and clears the screen. It works for files, messages, or text objects. The Abort instruction should be used, for example, when you realize you have asked for the wrong type of message form. It must be typed in.

FINISH retains all changes made, closes the object and clears the screen. It works for files, messages, or text objects. The Finish Instruction key would be used when you have completed working with an object. This is also done automatically by SIGMA when you ask for another object of the same type. But if you are only going to work on one object, a file for instance, you would use Finish when you are through with it.

SAVE retains changes made to a text object, but does not close the text object, leaving it on the screen for further work. This Instruction key would be used if you are going to leave the terminal for a moment, and you want to be sure your work on a text object up to that point is preserved.

Instruction key: !!SAVE!!

SHOW

The SHOW Instruction keys show the open object of each type (FILE, TEXT, or MESSAGE). They are used to get the open object back on the screen when the user has displayed something else instead. For example, after displaying a file, if the user displays an entry from that file and then wants to see the file again, he would use the !!SHOW-FILE!! key.

Instruction keys: !!SHOW-FILE!!

!!SHOW-MESSAGE!!

!!SHOW-TEXT!!

SORT

The SORT Instruction allows a user to reorder the entries in one of his files into Date Time Group (DTG) order, most recent DTG last. The file must belong to the user. If the file is open at the time the instruction is executed, it will be FINISHed.

Instruction Form: SORT (FILE) <EXISTING FILE NAME>

SYSTEM NEWS

The Typed Instruction SYSTEM NEWS is used to View the News which appears when you first log on. The News always appears when you log on and can be cleared with the !!CLEAR-VIEW!! key. To see the News again after having Logged on, you would type the SYSTEM NEWS instruction.

Instruction Form: SYSTEM NEWS

SYSTEM STATUS

The Typed Instruction SYSTEM STATUS is used to View the status of all SIGMA users who are active at a particular time.

Instruction Form: SYSTEM STATUS

UPDATE

When applied to messages the Update instruction key causes SIGMA to recognize and expand the Precedence field of the message and to check local addressees in the To, CC, Distr, Chop and Release fields. If you are not satisfied with this recognition and expansion, you may edit the fields accordingly.

Certain fields may appear in a different manner from the way in which they were originally typed. The name recognition is identical to what is done when a Coordinate or Release instruction is performed. You can use the Update instruction at any point to check on what SIGMA will do to the message before releasing or sending the message for chop. However, when you do a Coordinate or Release, the equivalent of an Update is done automatically, and you are asked to check recognition and acknowledge at that time.

Additionally the Update instruction causes the system to format the text of the displayed object (text object or message) according to the format rules built into SIGMA. Basically, individual lines are combined together, unless a line begins with a blank space. Lines beginning with a blank space are not formatted, but are left as typed.

In order to permit formatting of limited portions of text, Update accepts HERE markers as parameters. A single HERE causes SIGMA to format from the HERE to the end of the field (or object). If two HEREs precede the Update, SIGMA will reformat only the text between the HEREs.

Instruction key !!UPDATE!!

Instruction key <HERE>!!UPDATE!!

Instruction key <HERE><HERE>!!UPDATE!!

VIEW

The View instructions place information in the view window. The view window allows you to see file entries, messages or text objects, without being able to edit them. Thus, no Abort, Finish or Save instruction need be given since no changes may be made to any object in the view window. The CLEAR-VIEW instruction key erases the view window. Objects in the view window are indicated by being displayed in half intensity. The various instructions below allow the user to view selectors, messages, and text objects.

You cannot directly view a file. You must first DISPLAY the file, then use the !!VIEW-DISPLAY!! Instruction key. Note that if the file is a large one, the View window may not contain all of it after you press the View-Display key. In particular, if you scroll the View window, you may not be able to scroll to the ends of the file, even though you can do this in the Display window.

The first form is an Instruction key which shows in the View window a copy of whatever is being shown in the Display window.

Instruction key: !!VIEW-DISPLAY!!

The following form is typed in to view the named text object.

Instruction Form: VIEW TEXT <EXISTING TEXT NAME>

The following instruction form is typed. It views the Message IDentified by the Message ID.

Instruction Form: VIEW (MESSAGE) <Message ID>

The following Instruction keys view the next or current message in the open file. They require that there be an open file.

Instruction keys: !!VIEW-NEXT!!

!!VIEW-ENTRY!!

Similar to the above, the following Instruction key views the message selected in the open file.

Instruction key: <HERE> !!VIEW-ENTRY!!

The next form is typed in. It views the numbered entry in the open file.

Instruction Form: VIEW (ENTRY) <NUMBER>

The next instruction form is typed. It places a copy of the specified named selector in the view window.

Instruction Form: VIEW SELECTOR <EXISTING SELECTOR NAME>

The following typed instruction places a coordinator's version of the current message in the view window for reference.

Instruction Form: VIEW (VERSION) <USER ID>

The following typed instructions allow you to view a directory of files, text objects, or selectors which belong to some other user.

Instruction Forms: VIEW FILE DIRECTORY <USER ID>

VIEW SELECTOR DIRECTORY <USER ID>

Instruction Form: VIEW TEXT DIRECTORY <USER ID>

The following typed instruction presents in the View window all the keywords associated with the file being displayed.

Instruction Form: VIEW KEYWORDS