

**PUBLICATIONS
UPDATE**

Operating System/4 (OS/4)

Mini Message
Control Program

Programmer Reference

UP-8087-B

This Library Memo announces the release and availability of Updating Package B to "SPERRY UNIVAC Operating System/4 (OS/4) Mini Message Control Program Programmer Reference", UP-8087.

This update includes various corrections and additions to Section 2 and a small change to Section 4. The CONNECT=YES keyword parameter has been added to the ABC macro instruction.

Copies of Updating Package B are now available for requisitioning. Either the updating package alone, or the complete manual with the updating package may be requisitioned by your local sperry Univac Representative. To receive the updating package alone, order UP-8087-B. To receive the complete manual, order UP-8087.

LIBRARY MEMOS ONLY

Library Memo 217,
630, (less 631E, 634,
635A) 692, 60, 61, 65,
and 66

LIBRARY MEMO AND ATTACHMENTS

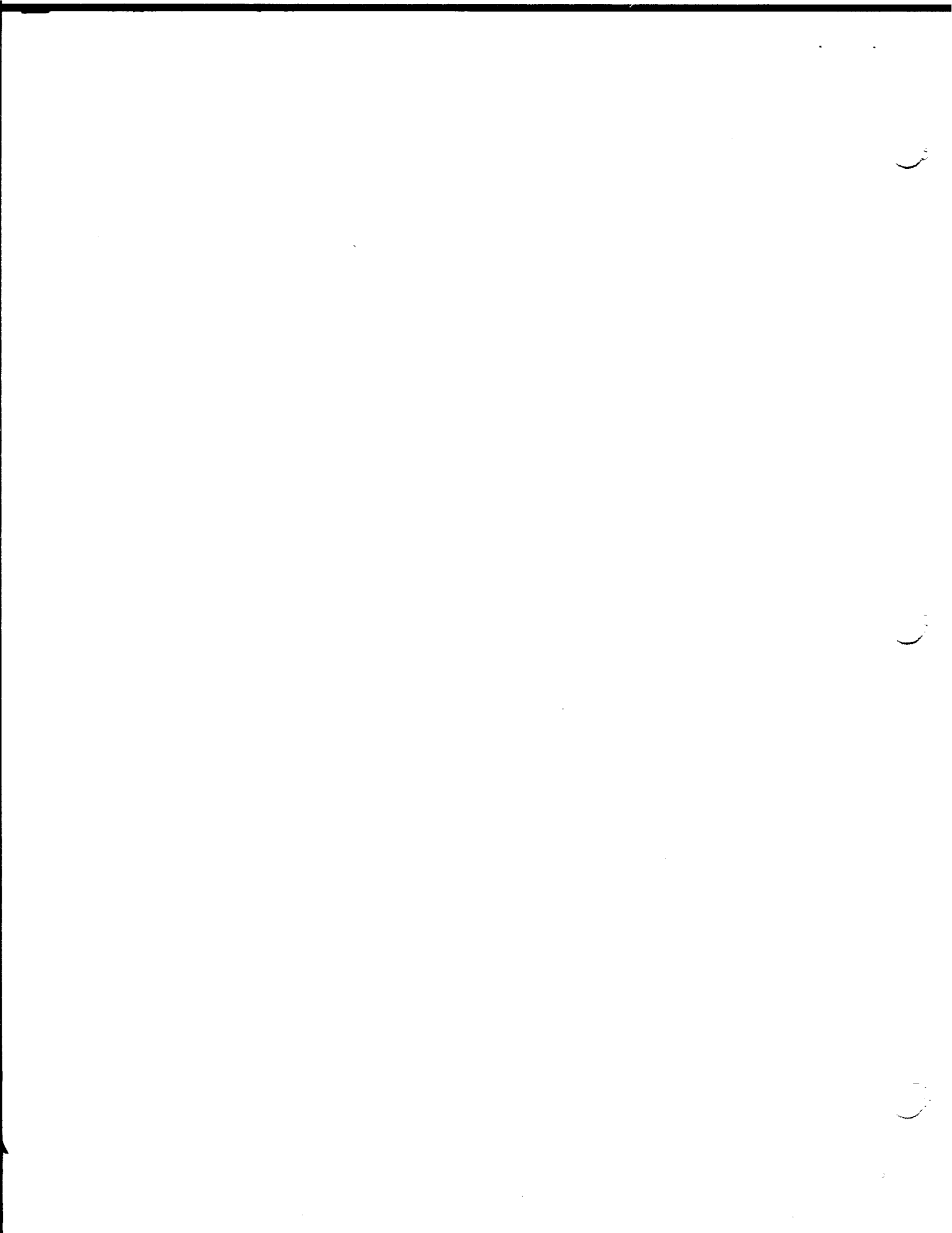
Mailing Lists 631E, 634, and 635A
(Package B to UP-8087, 19 pages plus Memo)

THIS SHEET IS

Library Memo for
UP-8087-B

RELEASE DATE:

March, 1976



UNIVAC OS/4 Mini Message Control Program

Programmer Reference

This document contains the latest information available at the time of publication. However, Sperry Univac reserves the right to modify or revise its contents. To ensure that you have the most recent information, contact your local Sperry Univac representative.

UNIVAC is a registered trademark of the Sperry Rand Corporation.

Other trademarks of the Sperry Rand Corporation include:

FASTRAND

MATED-FILM

PAGEWRITER

UNISCOPE

UNISERVO

SPERRY UNIVAC Operating System/4 (OS/4)

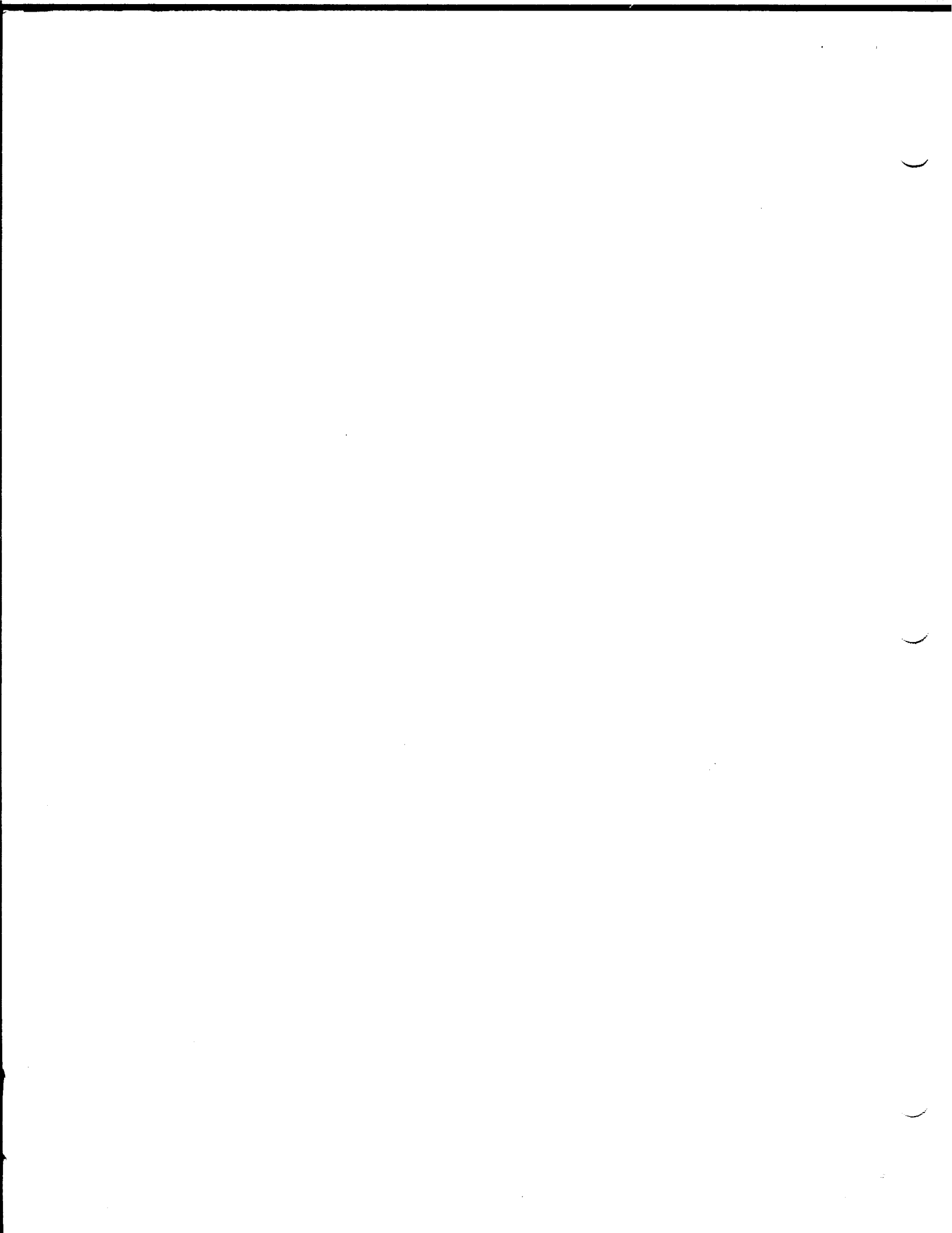
Mini Message Control Program

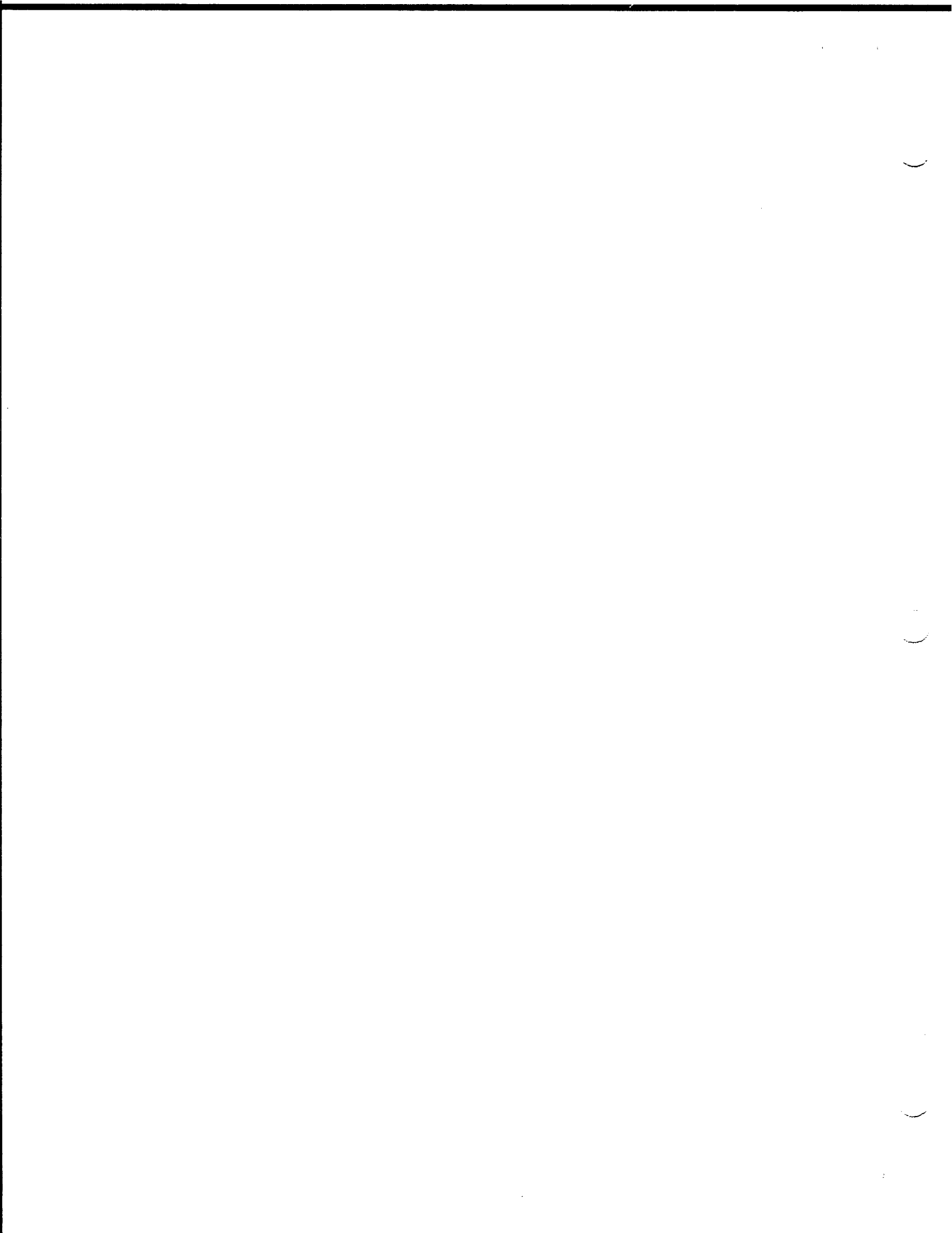
Programmer Reference

This document contains the latest information available at the time of publication. However, Sperry Univac reserves the right to modify or revise its contents. To ensure that you have the most recent information, contact your local Sperry Univac representative.

Sperry Univac is a division of Sperry Rand Corporation.

AccuScan, FASTRAND, PAGEWRITER, SPERRY UNIVAC, UNISCOPE, UNISERVO, and UNIVAC are trademarks of the Sperry Rand Corporation.





Contents

PAGE STATUS SUMMARY

CONTENTS

1. INTRODUCTION

1.1.	GENERAL DESCRIPTION	1-1
1.2.	STATEMENT CONVENTIONS	1-2
1.3.	DECLARATIVE MACRO INSTRUCTIONS	1-2
1.4.	IMPERATIVE MACRO INSTRUCTIONS	1-3
1.5.	COMMUNICATIONS APPLICATIONS	1-3
1.6.	PROGRAM DESCRIPTION	1-4
1.6.1.	Message Flow	1-4
1.6.2.	Error Processing	1-5

2. MMCP GENERATION AND USER INTERFACE ROUTINES

2.1.	DECLARATIVE MACRO INSTRUCTIONS	2-1
2.1.1.	MMCP Macro Instruction	2-1
2.1.1.1.	MMCP User Queueing Own Code (UQOC)	2-5
2.1.1.2.	MMCP Disc Queueing	2-6
2.1.2.	ABC Macro Instruction	2-7
2.1.3.	DTLNE Macro Instruction	2-8
2.1.4.	MMTRM Macro Instruction	2-9
2.1.5.	MMCP Assembly, Linking, and Running	2-11
2.2.	IMPERATIVE MACRO INSTRUCTIONS	2-12
2.2.1.	Line Open Macro Instruction (OPNMM)	2-12
2.2.1.1.	User Error Interface	2-14
2.2.2.	Obtain Input Message Macro Instruction (GETMM)	2-16
2.2.3.	Send Output Message Macro Instruction (PUTMM)	2-18
2.2.4.	Line Close Macro Instruction (CLSMM)	2-22

3. SUPERVISOR GENERATION

4. OPERATOR COMMUNICATIONS

4.1.	UNSOLICITED TYPE-INS	4-1
4.2.	MMCP TYPE-OUTS	4-2

APPENDIX

A. SYMBOLIC DEVICE IDENTIFIERS

FIGURES

1-1.	General Message Flow	1-4
2-1.	Message Received from GETMM	2-17
2-2.	Message Sent by PUTMM	2-20

TABLES

2-1.	User Error Information Block	2-15
2-2.	Flag Bytes in Error Information Block	2-15

1. Introduction

1.1. GENERAL DESCRIPTION

The mini message control program (MMCP) is a compact software communications package which provides the UNIVAC OS/4 Operating System user with the ability to generate message control and message processing programs. These programs control all the input/output data transfers over communications lines. The following terminal devices may be included in the communications network:

UNISCOPE 100 Display Terminal (UNISCOPE 100)

UNIVAC DCT 1000 Data Communications Terminal (DCT 1000)

Communications control consists of three essential components: the communications physical input/output control system (CPIOCS), the mini message control program (MMCP), and the user's message processing program (only one user application program may be interfaced with MMCP). CPIOCS is a facet of the system resident supervisor and includes the I/O dispatcher and interrupt handling routines.

Macro instructions necessary and/or peculiar to MMCP are provided to construct a complete program that controls the movement of messages to and from a variety of commonly used remote terminals and the user message processing program. Messages in text are received from the remote terminals and linked to an input queue. From the input queue, the messages are transferred, as needed, for use by the message processing program.

Since the MMCP is generated by macro instructions, the user can specify the communications equipment configuration and the number and size of buffer areas to be used in both main storage and direct access storage.

Through line and terminal definition and control macro instructions, the program builds the links and tables that control the operation of the communications processing. In effect, the remote terminals become part of the input/output system of the central processor and are accessed just as other input/output devices. The programmer does not have to write assembly level statements but he must understand the function of these terminals and the flow of data generated by the program coding.

A knowledge of the *UNIVAC 9400/9480 Systems OS/4 Supervisor Programmer Reference, UP-7689* (current version), *UNIVAC OS/4 Data Management System Programmer Reference, UP-7629* (current version), *UNIVAC 9400 System Assembler/Central Processor Unit Programmer Reference, UP-7600* (current version), or *UNIVAC 9700 System OS/4 Assembler Programmer Reference, UP-7935* (current version), and *UNIVAC OS/4 Communications Control Program Programmer Reference, UP-7808* (current version) is helpful in using this manual.

1.2. STATEMENT CONVENTIONS

The conventions used to illustrate the statements in this manual are as follows:

- Capital letters and punctuation marks (except braces, brackets, and ellipses) are information that must be coded exactly as shown.
- Lowercase letters and terms represent information that must be supplied by the programmer.
- Information contained within braces represents necessary entries of which one must be chosen.
- Information contained within brackets represents optional entries that (depending on program requirements) are included or omitted. Braces within brackets signify that one of the entries must be chosen if that operand is included.
- An ellipsis (three spaced periods) indicates the presence of a variable number of entries.
- Commas are required after each parameter except after the last parameter specified. When a positional parameter is omitted from within a series of parameters, the comma must be retained to indicate the omission.

1.3. DECLARATIVE MACRO INSTRUCTIONS

A message processing program must inform the message control program of the parameters, special conditions, current status, and options pertaining to a file, line, or terminal. This information is included in a declarative macro instruction for each file, line, or terminal required by the message processing program. The declarative macro instruction and the selected keyword parameters in the operand field define the file, line, or terminal name. A keyword parameter consists of a word or code and an equal sign, followed by one specification.

The format of the declarative macro instruction is:

LABEL	△ OPERATION △	OPERAND
[name]	code	keyword ₁ =x[,keyword ₂ =y] ... [,keyword _n =z]

The symbolic name of the file, line, or terminal must appear in the label field. This name can have one to four alphanumeric characters and must begin with an alphabetic character. The appropriate code must appear in the operation field. The keyword parameters can be written in any order in the operand field and must be separated by commas. Appropriate assembler rules regarding macro instructions apply to blank columns and continuation.

The alternate form of writing the declarative macro instructions is:

LABEL	△ OPERATION △	OPERAND	COLUMN 72
[name]	code	keyword-1=x, keyword-2=y, . . . keyword-n=z	X X . . .

In the alternate form, a continuation mark is necessary in column 72 of every line except the last. Each keyword parameter and specification except the last must be followed by a comma.

1.4. IMPERATIVE MACRO INSTRUCTIONS

A message processing program must be able to communicate with the message control program to process files, lines, or terminals that have been defined by declarative macro instructions. This processing is accomplished by including imperative macro instructions in the message processing program.

The format of the imperative macro instruction is:

LABEL	△ OPERATION △	OPERAND
[name]	XXXX	YYYY,...,ZZZZ

A symbolic name may appear in the label field. The name can have a maximum of eight alphanumeric characters and must begin with an alphabetic character. The appropriate verb or code must appear in the operation field. The positional parameters (as signified by a name or register number) must be written in the specified order in the operand field and must be separated by commas. When a positional parameter is omitted, the comma must be retained to indicate the omission, except in the case of omitted trailing parameters. Appropriate assembler rules regarding macro instructions apply to blank columns and continuation.

1.5. COMMUNICATIONS APPLICATIONS

The MMCP can handle a variety of communications applications, which fall into two general categories: message control and message processing. Message control applications involve switching input messages to various specified destinations, and collecting data for further processing by a user-generated message processing program.

Message processing applications involve the processing of messages by a user-generated message processing program. A few message processing applications are:

- Processing inquiries
- Processing collected data
- Updating inquiry files
- Updating message routing files
- Processing management information

An inquiry application involves receiving messages from terminals (performed by the MMCP), processing the data contained in the messages (performed by the user's message processing program), and sending replies to the originating terminals (MMCP).

The processing of collected data is the second part of a 2-step application. The first step is the actual collection of the data by (MMCP), which places the collected data in the direct access storage device process queues. These messages may be retained on the storage device used until it is convenient to process them; that is, they remain on the queue until a message processing program issues GET macro instructions to obtain and process the messages.

1.6. PROGRAM DESCRIPTION

MMCP consists of a number of modules which interface with the supervisor and the user's program. MMCP, assigned problem program priority 1, is responsible for initiating and controlling message flow.

1.6.1. Message Flow

If the terminal and line are up (eligible to be polled) and the terminal is not within its poll interval (wait state), the line handler continuously polls remote terminals for input, passing control to the supervisor dispatcher routine which sends the poll. See Figure 1-1.

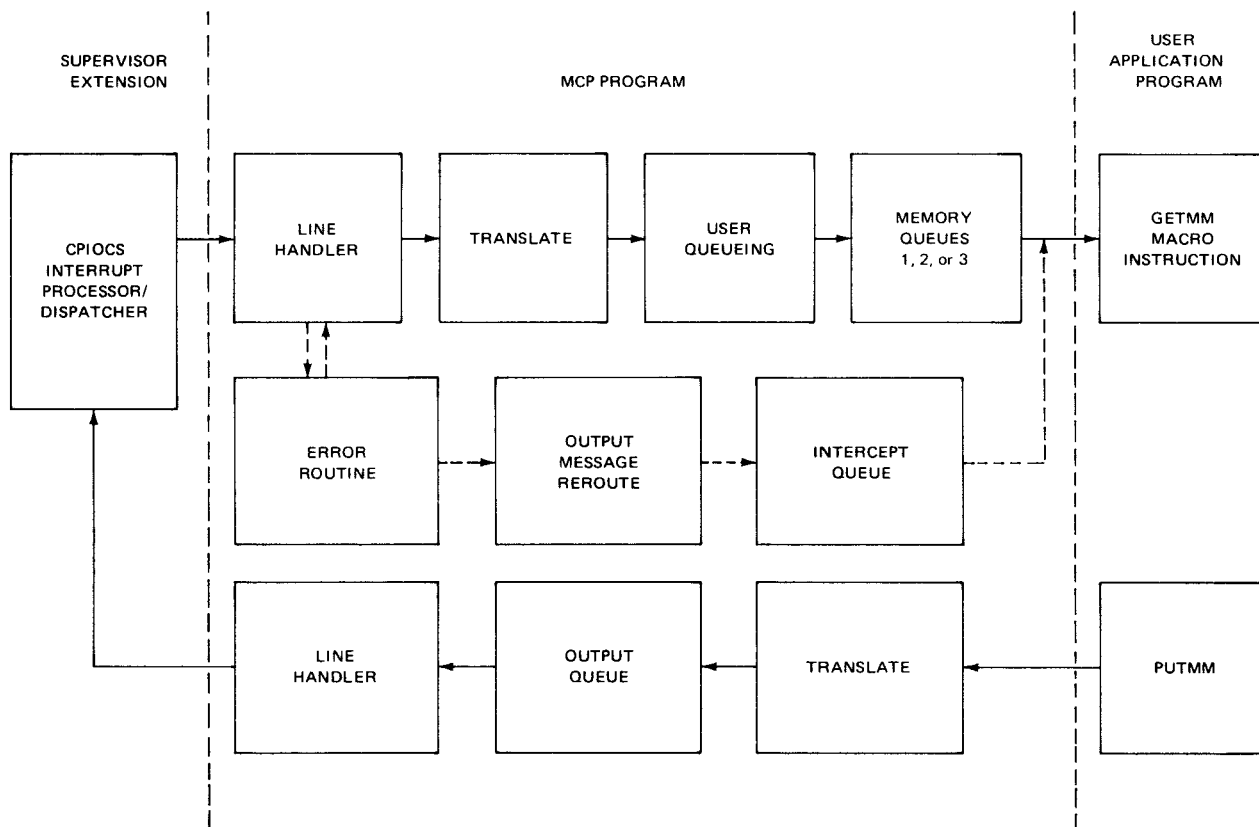


Figure 1-1. General Message Flow

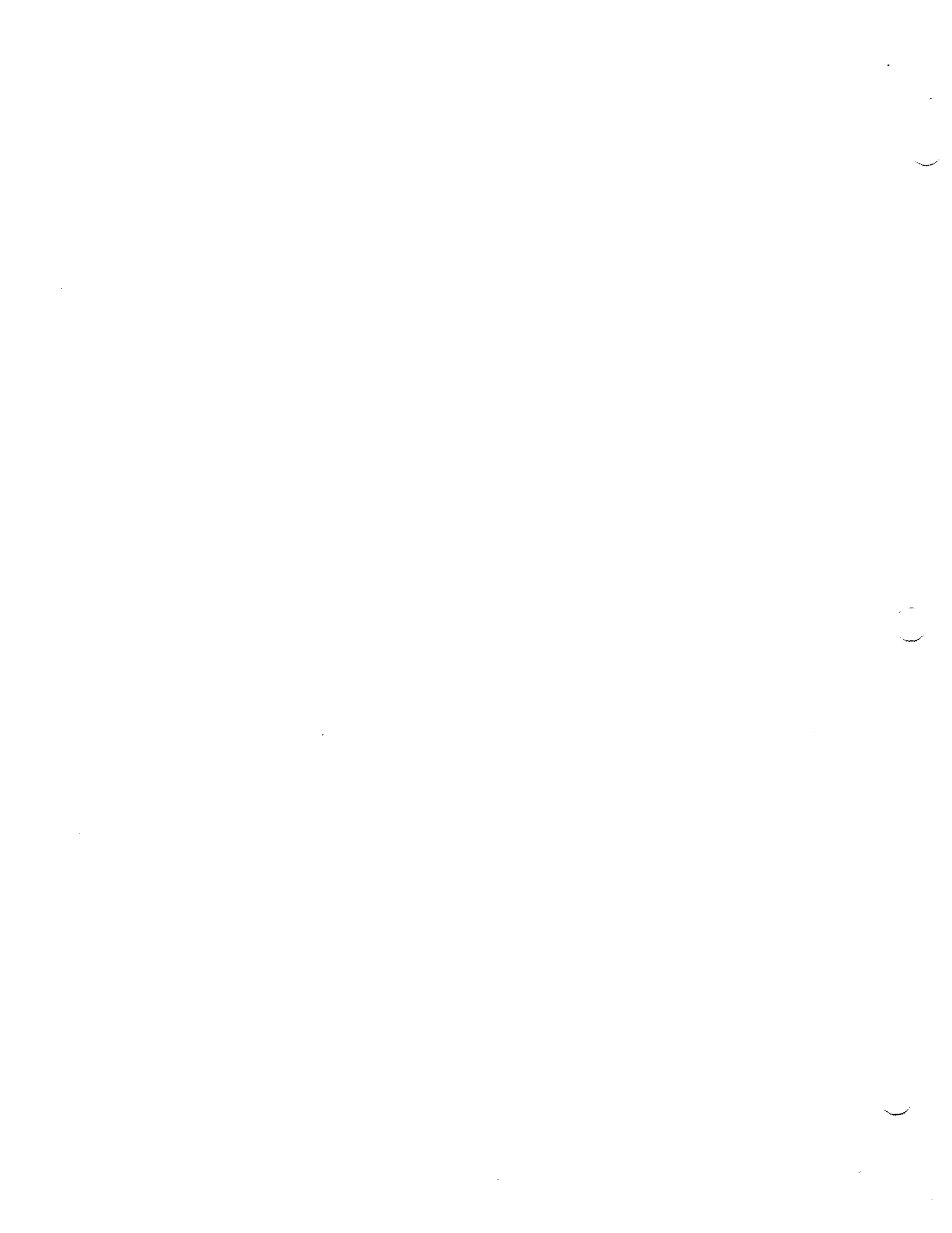
The reply from the terminal is stored in a buffer or a control message area. On return to the line handler, the response is investigated. If the reply is a 'no traffic' response, the handler passes on to the next terminal or surrenders control to the central control routine which starts processing for the next line. However, if the response is text, the message is translated and the terminal identification and the message are linked to an input queue. If more than one queue has been specified in the MMCP generation, the user must determine the input queue to which the message is to be linked via the user queueing own code (UQOC) routine (2.1.1.1).

When the user issues a GETMM command, the first message is transferred from the required input queue to the user's work area. The user should note that, if intercept queue facilities have been specified, a GETMM command will result in an intercept message if the intercept queue is in use.

When the user issues a PUTMM command, the message is translated, the symbolic terminal identification from the terminal list is replaced by the physical identification and inserted in the message, and the message is placed on the end of the relevant output queue. During the normal line handler sequence, the output queue is continually scanned for output. If the output queue is not empty, the first message on the output queue is dispatched to the relevant terminal.

1.6.2. Error Processing

If an error is detected during processing (i.e., the terminal or line is found not to be active or an expected response is not received) MMCP will attempt to recover from the error by repeating the last sequence of events a maximum of five times. If the error continues, control is passed to the MMCP error routine which analyses the error conditions and takes the necessary action (i.e., setting the terminal down). The console operator is informed of the error and may take recovery action. The user program will also be informed of the error if an error address has been defined for the line in question (2.2.1.1).



2. MMCP Generation and User Interface Routines

2.1. DECLARATIVE MACRO INSTRUCTIONS

The mini message control program (MMCP) is generated using four declarative macro instructions: MMCP, ABC, DTLNE, and MMTRM. Keywords of these instructions can be varied to build a program to the user's requirements. If two or three queues are specified, a user queueing own code (UQOC) routine must be included in the assembly.

2.1.1. MMCP Macro Instruction

This instruction defines specifications relative to the user's system requirements. The format of the MMCP instruction is:

LABEL	Δ OPERATION Δ	OPERAND
name	MMCP	[BUFSIZE=n] [CYL=n] [DISC={ 8411 8414 8424 }] [D1000=n] [U100=n] [ERROR={ YES } { NO }] [INTERCPT={ YES } { NO }] [LINES=n] [NUMBUF=n] [NUMQ=n] [NUMTRM=n] [POLLINT=n] [QUEUE=DISC] [UQOC=label]

where:

name
Is a user-assigned label.

NOTE:

Each keyword parameter and specification used must be separated by a comma. If more than one line of specifications is used, all lines except the last must contain a continuation mark in column 72.

The following explanations describe specific keyword parameters of the MMCP macro instruction:

- Buffer Size

This optional keyword parameter specifies the fixed buffer size. The format is:

BUFSZE=n

where:

n

Equals the number of bytes required to hold the maximum length user message, which consists of a 2-byte message length field, a start of header character (SOH), a 3-byte terminal identifier, and the characters start of text (STX) through end of text (ETX). If BUFSZE is not specified, a buffer size of 100 bytes will be assumed.

- Number of cylinders

This optional keyword parameter specifies the number of cylinders required for the disc buffer when disc queueing is used (2.1.1.2). The format is:

CYL=n

where:

n

Equals the number of cylinders required.

- Type of disc

This optional keyword parameter specifies the type of disc pack to be used for the buffer area when disc queueing is used (2.1.1.2). The format is:

DISC= $\left\{ \begin{array}{l} 8411 \\ 8414 \\ 8424 \end{array} \right\}$

NOTE:



To select the 8425 disc pack, the 8424 parameter should be specified.

- Device type

The types of terminal devices included in the system are defined by one or more of the following parameters:

U100=n

D1000=n

which define the existence in the system of UNISCOPE 100 Display Terminals and UNIVAC DCT 1000 Data Communications Terminals, respectively.

In each case, n is the time-out in seconds required for the particular device type. The time-out time should be at least as long as the time required to transmit the maximum length message (e.g., if the maximum length message is 1000 bytes, and the line speed is 1200 baud, the time required to transmit the message is approximately 6.6 seconds and the specified time-out time must therefore be at least 7 seconds).

- Error interface

This optional keyword parameter must be specified as YES if the user wishes to make use of the error exit address facility in the OPNMM macro instruction (2.2.1) and the resultant error processing interface (2.2.1.1). The format is:

$$\text{ERROR} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$$

If YES is specified, the exit is generated. If this parameter is omitted, ERROR=NO is assumed.

- Intercept queueing

This optional keyword parameter specifies intercept queueing. The format is:

$$\text{INTERCPT} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$$

where:

INTERCPT=YES

Specifies messages in an output queue associated with a malfunctioning (downed) terminal or line are placed on the intercept queue and returned to the user on execution of a GETMM macro instruction.

INTERCPT=NO

Specifies messages in the output queues of downed terminals or lines are discarded.

If omitted, INTERCPT=YES is assumed.

- Number of lines

This optional keyword parameter specifies the total number of lines available to the system. The format is:

LINES=n

where:

n

Equals the number of lines defined in the system and is less than or equal to 64.

If not specified, one line is assumed.

- Number of buffers

The optional keyword parameter specifies the total number of buffers to be generated. The format is:

NUMBUF=n

where:

n

Equals the total number of buffers and must be at least equal to the number of lines specified in the LINES keyword parameter. The minimum number of buffers that may be generated is 2.

If omitted, the number of buffers is computed according to the job storage limits at run time. This allows the user to assign buffer area at run time.

- **Number of queues**

This optional keyword parameter specifies the number of input queues. The format is:

NUMQ=n

where:

n

Is the integer 1, 2, or 3. If more than one queue is specified, a user own code routine should also be included in the generation of MMCP to determine the queue to which input messages are to be routed (2.1.1.1).

If omitted, NUMQ=1 is assumed.

- **Number of terminals**

This optional keyword parameter specifies the number of terminals in the system. The format is:

NUMTRM=n

where:

n

Equals the number of terminals defined in the system.

If omitted, one terminal is assumed.

- **Poll interval**

This optional keyword parameter specifies the interval between polling passes. The format is:

POLLINT=n

where:

n

Equals the number of milliseconds between polling passes.

If omitted, an interval of zero is assumed.

- **Type of queueing**

This optional keyword parameter specifies the type of queueing to be run. When disc queueing is to be performed this keyword parameter must be specified. See 2.1.1.2. The format is:

QUEUE=DISC

If omitted, memory queueing is assumed.

- User own code routine

This keyword parameter specifies the address of the user own code routine. The format is:

UQOC=label

where:

label

Is that of the user routine (2.1.1.1). This parameter should not be specified if the parameter NUMQ is omitted or is equal to 1.

2.1.1.1. MMCP User Queueing Own Code (UQOC)

If UQOC=label is specified on the MMCP generation, a user queueing own code module must be assembled with MMCP. This routine is entered when an input message has been received and a queue has to be chosen for linking the message. The user is given the address of the translated message in register 1; however, register 0 should first be checked to ensure the translation was successful. On exit, the user should put the required queue number in register 0. The user may not execute any of the following functions in the UQOC:

- Data management macro instructions
- STXIT macro instruction
- Physical I/O
- Any function that relinquishes control
- Transient calls
- Message control macro instructions

The following entry and exit conditions should be observed:

- Entry conditions
 - Register 0
 - Is 1 if translation is unsuccessful; i.e., start of text or start of entry character is missing.
 - Register 1
 - Is the address of the message which has the same format as that presented to the user in the GETMM.
 - Register D
 - Is not to be corrupted.
 - Register E
 - Is the return address to MMCP.
 - Register F
 - Is the address of UQOC routine.

■ Exit conditions

— Register 0

Is the number of the input queue.

— Register 1

Is the address of message.

— Register D

Is the register D as an entry.

Exit must be through register E.

Example:

LABEL	OPERATION	OPERAND	COMMENTS
1	ENTRY	QUEUQUIZ	
	USING	QUEUQUIZ,RF\$	BASE REGISTER
QUEUQUIZ	LTR	RO\$,RO\$	MSG. TRANSLATED?
	BNZ	QUEUO2	NO
	CLI	14(RI\$),C'1'	FOR QUEUE 1?
	BNE	QUEUO2	NO
	LA	RO\$,1(,RO\$)	SET QUEUE 1
	B	QUEUEND	EXIT
QUEUO2	LA	RO\$,2(,RO\$)	SET QUEUE 2
QUEUEND	BR	RE\$	EXIT

2.1.1.2. MMCP Disc Queuing

MMCP may be generated with disc buffers as well as main storage buffers. Disc space may be obtained on the disc pack, either on 8411, 8414, 8424, or 8425. Disc queuing utilizes temporary disc storage space (SYSPOOL); however, the entire 200 cylinders of the 8411 or 8414 discs, or the first 220 cylinders of the 8424 or 8425 discs may be used if required. When writing to input or output queues, main storage buffers are used in preference to disc buffers if more than three buffers are unallocated at the time. Thus, disc buffers are only utilized in an overflow situation.

To generate MMCP with disc queuing the following parameters must be included in the MMCP macro instruction:

QUEUE=DISC

DISC= { 8411
8414
8424 }

CYL=n

NOTE:

To select the 8425 disc pack, the 8424 parameter should be specified.

where:

n
Equals the number of cylinders required.

The preceding keyword parameters will generate full disc processing capability plus a track bit map. The size of the track bit map depends on the number of cylinders specified; i.e., if an 8414 disc with 200 cylinders has been specified, the bit map will occupy 500 bytes.

MMCP with disc queueing is run similarly to main storage queueing MMCP except that the disc with SYSPool must be specified in the job control as follows:

```
// DVC      10
// VOL      DSP004
// LFD      SYSPool
```

Before MMCP prints out the initial message, the disc will be formatted. Time taken for this formatting will depend on the number of cylinders specified.

2.1.2. ABC Macro Instruction

This instruction generates the remainder of the mini MCP modules. The format of the ABC instruction is:

LABEL	△ OPERATION △	OPERAND
[name]	ABC	[QUEUE=DISC] [,CONNECT=YES]

where:

name
Is a user-defined label.

QUEUE=DISC
Specifies the type of queueing to be run. This optional keyword parameter must be used when QUEUE=DISC is specified in the MMCP macro instruction.

If omitted, memory queueing is assumed.

CONNECT=YES
Must be specified if a terminal connection cannot be made before the data communications subsystem is activated by the execution of the MMCP job. Normally, the MMCP assumes that all terminal connections have been established before the MMCP is executed. The use of this optional parameter delays the activation of the input line for one minute to allow terminal connection to be established.

2.1.3. DTLNE Macro Instruction

This instruction is used to generate a definition table for each line in the system. All lines must be defined consecutively and in line poll sequence. The format is:

LABEL	Δ OPERATION Δ	OPERAND
→ linename	DTLNE	INLINE=n POLLST=(term ₁ , ..., term _n) [DEVICE=U1/D1] [DIRINT=YES] [DUPLICATE=F]

where:

linename

Is the unique 3-character symbolic line name assigned by the user.

NOTE:

Each keyword parameter and specification used must be separated by a comma. If more than one line of specifications is used, all lines except the last must contain a continuation mark in column 72.

Keyword parameters of the DTLNE instruction are as follows:

- Line identifier

This keyword parameter defines the line address. The format is:

INLINE=n

where:

n

Is the physical input line address and must be an odd number between 1 and 127. The output line address is assumed to be n minus 1.

NOTE:

No two DTLNE macro instructions may specify the same value for n.

- Poll list

This keyword parameter specifies the poll list for the line. The format is:

POLLST=(term₁, term₂, . . . , term_n)

where:

term₁, term₂, . . . , term_n

Are the symbolic terminal names as defined in the label field of the MMTRM macro instruction in the order in which the terminals are to be polled. A terminal may be specified more than once.

- Device type

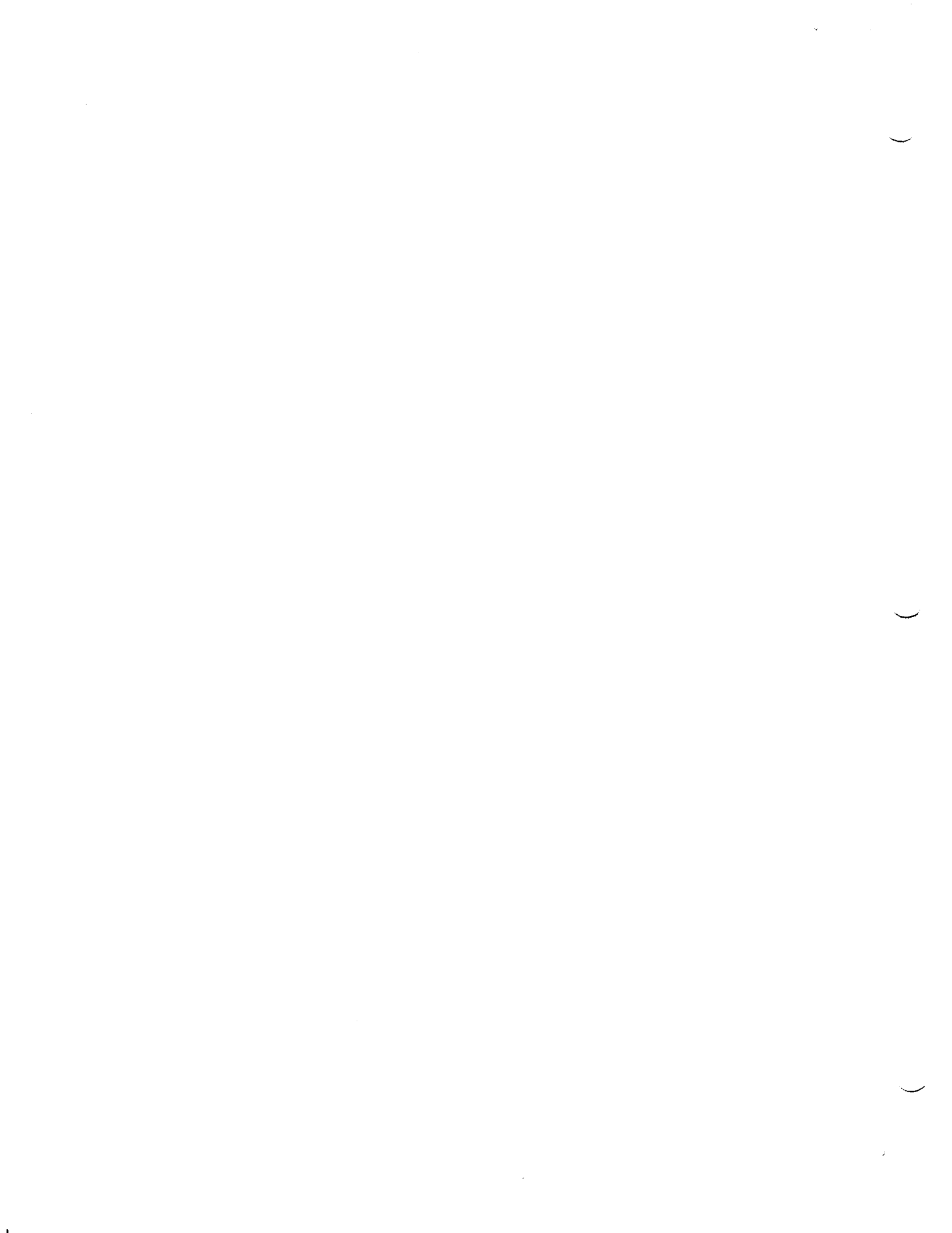
This optional keyword parameter defines the device type for the line. The format is:

DEVICE=U1/D1

where:

U1/D1

Defines the UNISCOPE 100 and/or DCT1000.



■ Direct interface

This optional keyword parameter specifies that a line is on a direct interface. The format is:

$DIRINT = \begin{cases} YES \\ NO \end{cases}$

If this parameter is omitted, DIRINT=NO is assumed.

■ Duplex lines

This optional keyword parameter specifies that a line is using full duplex transmission and has the format:

DUPLEX=F

where:

DUPLEX=F

Must be specified for direct interface connections, or if high speed modems are used. ←

If omitted, a half duplex line is assumed.

2.1.4. MMTRM Macro Instruction

This macro instruction is used to generate a definition table for each terminal available to the system. All terminals must be defined consecutively, grouped by line. The format is:

LABEL	Δ OPERATION Δ	OPERAND
terminal-name	MMTRM	ADDR=nnnn DEVICE= { U100 } { D100 } LINE=symbol POLL=nnnn

where terminal-name is the unique 2-character user-defined symbolic terminal identifier. All references to the terminal by the user are via these 2 characters together with a device-dependent third character identifying the device on the remote terminal to which the message is to be routed or from which the message was received; e.g., the character S identifies the screen, and the character P identifies the incremental printer for the UNISCOPE 100 (Appendix A).

NOTE:

Each keyword parameter and specification used must be separated by a comma. If more than one line of specifications is used, all lines except the last must contain a continuation mark in column 72.

Keyword parameters of the MMTRM instruction are as follows:

- Addressing characters

This keyword parameter specifies terminal addressing, rid, or sid. The format is:

→ **ADDR=nnnn**

where:

→ **nnnn**
Is a hexadecimal number.

- Device type

This keyword parameter specifies the terminal type being defined by the MMTRM macro instruction. The format is:

**DEVICE= { U100 }
 { D1000 }**

where:

U100 and D1000
Define a UNISCOPE 100 and DCT1000, respectively.

All terminals must be defined together immediately following the last line definition. Terminals must be defined in line groups in the same order as the lines previously defined. Within line groups, all terminals with the same polling characters must be defined together.

- Line name

This keyword parameter identifies the line to which the terminal is attached. The format is:

LINE=symbol

where:

symbol
Is the label of the associated DTLNE macro instruction.

- Polling characters

This keyword parameter specifies the terminal poll, remote identifier/general identifier (rid/gid), or station identifier/general identifier (sid/gid). The format is:

→ **POLL=nnnn**

where:

→ **nnnn**
Is a hexadecimal number.

2.1.5. MMCP Assembly, Linking, and Running

After updating the procedure library from the release tape, MMCP is generated with one assembler run and one linkage editor run, as shown in the following example. The series of instructions in the example illustrates the coding required to generate MMCP for a system consisting of three lines and ten UNISCOPE 100 terminals using two input queues.

Example:

LABEL	OPERATION	OPERAND	COMMENTS
1	10	15	72
COMMS	START	O	
	STDEQU	SB,JB,JP,MM	
MINI	MMCP	LINES=3,NUMBUF=10,BUF SIZE=200, POLLINT=500,NUMTRM=10,INTERCPT=NO, NUMQ=2,DEVICE=U100,QUEUQUIZ	X X
LN1	DTLNE	INLINE=01,POLLST=(T1,T4)	
LN2	DTLNE	INLINE=03,POLLST=(S1,S2,S3)	
LN3	DTLNE	INLINE=05,POLLST=(U1),DIRINT=YES	
T1	MMTRM	LINE=LN1,ADDR=2251,POLL=2250,DEVICE=U100	
T2	MMTRM	LINE=LN1,ADDR=2252,POLL=2250,DEVICE=U100	
T3	MMTRM	LINE=LN1,ADDR=2253,POLL=2250,DEVICE=U100	
T4	MMTRM	LINE=LN1,ADDR=2551,POLL=2550,DEVICE=U100	
T5	MMTRM	LINE=LN1,ADDR=2552,POLL=2550,DEVICE=U100	
S1	MMTRM	LINE=LN2,ADDR=2351,POLL=2351,DEVICE=U100	
S2	MMTRM	LINE=LN2,ADDR=2451,POLL=2451,DEVICE=U100	
S3	MMTRM	LINE=LN2,ADDR=2452,POLL=2452,DEVICE=U100	
U1	MMTRM	LINE=LN3,ADDR=2651,POLL=2650,DEVICE=U100	
U2	MMTRM	LINE=LN3,ADDR=2652,POLL=2650,DEVICE=U100	
	ENTRY	QUEUQUIZ	
	USING	QUEUQUIZ,RF\$	
QUEUQUIZ	LTR	RO\$,RO\$	
	BNZ	QUEUO2	
	CLI	14(RI\$),C'1'	
	BNE	QUEUO2	
	LA	RO\$,1(,RO\$)	
	B	QUEUEND	
QUEUO2	LA	RO\$,2(,RO\$)	
QUEUEND	BR	RE\$	
	ABC		
	END	COMMS	

The macro calls should be specified in the following order:

1. MMCP call
2. All DTLNE calls
3. All MMTRM calls in line order and grouped by poll characters; i.e., all terminals on a line having the same poll characters should be specified consecutively.
4. UQOC coding, if specified
5. ABC call

The module COMMS described in the preceding example, may be assembled to produce relocatable code and then linked, or it may be assembled to produce loadable code. The ABC macro call may be assembled separately and then linked, but all the other macro calls must be assembled together.

When running the communication system, MMCP should be loaded before the user application program and must be run at priority 1 in supervisor mode. The priority may be specified on the JOB card or the RUN type-in, and supervisor mode is achieved by including // OPTION SUPVMODE in the MMCP job stream. The user application may then be loaded and run.

2.2. IMPERATIVE MACRO INSTRUCTIONS

The user programmer has available four imperative macro instructions which permit the construction of programs for controlling a communications system. Using these four macro instructions and standard assembler instructions, the user programmer may develop a comprehensive message processing program to interface with MMCP. The macro instructions are:

1. OPNMM Line open
2. GETMM Obtain input message
3. PUTMM Send output message
4. CLSMM Line close

To enhance the message processing program further, the user has available an optional error processing interface to which control is passed by MMCP when an error condition has occurred and full normal recovery procedures have been exhausted.

2.2.1. Line Open Macro Instruction (OPNMM)

The line open macro instruction is used to initiate polling, and hence, must be executed prior to a GETMM or PUTMM macro instruction. On return from the line open, it is the responsibility of the user program to check the error half word generated by the expansion of the macro instruction. Only one communications line may be specified as a parameter in the operand field of this instruction. If an error is detected during processing, the line is not opened. The formats of the OPNMM macro instruction are as follows:

LABEL	△ OPERATION △	OPERAND
[name]	OPNMM	{ linename[,error-address] } { (1) }

where:

name

Is a user-defined label.

■ Positional Parameter 1:

linename

Is the 3-character symbolic name of the relevant DTLNE line macro instruction.

(1)

Register 1 is preloaded with the address of a packet in the format:

CNOP	2,4	Alignment
DC	H'0'	Error half word
DC	X'1'	Request identifier
DC	CL3'linename'	
DC	A(error-address)	

■ Positional parameter 2:

error address

Is the error address, an optional parameter giving the address of the user error processing routine. If the error parameter is omitted, the user program will not be aware of subsequent hardware errors.

The line open macro instruction produces the following expansion:

LABEL	Δ OPERATION Δ	OPERAND	COMMENTS
*LAB1	OPNMM	LN1,ER1	
	CNOP	2,4	adjust to word boundary +2
LAB1	BAL	R1\$,*+14	
	DC	H'0'	error half word
	DC	X'1'	line open indicator
	DC	CL3'LN1'	line name
	DC	A(ER1)	error address
	SVC	SV\$UPR	

Specifying the operand (1) generates the SVC instruction.

2.2.1.1. User Error Interface

During the processing of the line open macro instruction, bits are set in the error half word if the line open function has not been completed successfully. The error half word identifies the reason for failure, and should be checked following the execution of the OPNMM macro instruction. Bits may be set in the error half word to reflect the following hexadecimal configurations identifying the reasons for failure:

Reason	Half-word format
Request unknown	X'8000'
Second user attempting interface with MMCP	X'4000'
Invalid line-name	X'2000'
Line already open	X'1000'
MMCP not in main storage	X'0100'
MSG too short (minimum length message must be 4 bytes)	X'0080'
All buffers assigned to input	X'0040'

The user error interface routine may be defined as positional parameter 2 of the line open macro instruction. It is an optional routine which the user may include in his program to maintain a record of hardware errors. This routine is entered only when an error occurs or line up/down or terminal up/down type-in is received. The following conditions should be adhered to, to ensure system continuation:

■ Entry conditions

- Register 0 Reserved for MMCP
- Register 1 Address of the error information block
- Register 2 User registers restored
- Register 13 Reserved for MMCP
- Register 14 MMCP return address
- Register 15 Address of user routine entry

■ Exit conditions

The user must exit via register 14 with registers 0 and 1 the same as at entry. Exit to the user error processing routine is designed to make the user aware of the error only if normal error recovery has been attempted and failed. No facility is provided for further attempts.

The user must exit to message control and may not execute any of the following functions in the error routine:

- Data management macro instructions
- STXIT macro instruction
- Physical I/O

- Any function that relinquishes control
- Transient calls
- Message control macro instructions

Table 2-1 describes the user error information block. Table 2-2 lists bit settings of flag bytes in the error information block and their significance.

Table 2-1. User Error Information Block

Classification	Label (Displacement From R1\$)	Boundary Alignment	Field Length Bytes	Label Description
Line name	BU\$UE1	Full word	3	User's symbolic linename
Terminal name	BU\$UE2	Byte	3	User's symbolic terminal name and device identifier
Address	BU\$UE3	Byte	3	Terminal address/polling characters - rid/sid/did
Status	BU\$UE4	Byte	1	Device status
Sense	BU\$UE5	Half word	2	DCS sense bytes
Message	BU\$UE6	Full word	4	Address of message
Flag bytes	BU\$UE7	Half word	2	(See Table 2-2)

NOTE:

The letters rid/sid/did represent remote identifier, station identifier, and device identifier, respectively.

Table 2-2. Flag Bytes in Error Information Block

Bit Setting	Reason	Bit Setting	Reason
0 = 0	Input error	8 = 1	Terminal down
0 = 1	Output error	9 = 1	Invalid rid/sid/did
1 = 1	Start I/O (SIO) command failure	10 = 1	Unknown input response
2 = 1	I/O time-out	11 = 1	Expected ACK response not received
3 = 1	Bad message status	12 = 1	Redundant ACK response
4 = 1	Terminal busy time-out	13 = 1	Redundant not busy
5 = 1	Line up	14 = 1	Continuous reply request
6 = 1	Line down	15 = 1	Poll disabled
7 = 1	Terminal up		

2.2.2. Obtain Input Message Macro Instruction (GETMM)

The GETMM macro instruction is used to obtain a complete message from the message control program. The formats of the GETMM instruction are as follows:

LABEL	Δ OPERATION Δ	OPERAND
[name]	GETMM	work-area-name[,queue-number] [,no-message-available-address] (1)

where:

name

Is a user-defined label.

■ Positional parameter 1:

work-area-name

Specifies the label of the user's work area into which the message is to be placed.

(1)

The user preloads register 1 with the address of a packet in the following format:

CNOP	2,4	Alignment
DC	H'0'	Error half word
DC	X'3'	Request identifier
DC	AL3 (work-area-name)	
DC	A (no-message-available-address)	
DC	H'queue-number'	

■ Positional parameter 2:

queue-number

The queue number is an integer (1, 2, or 3) which specifies the input queue from which the message is to be obtained.

If queue 2 or 3 is specified, the user must have included an own code module in the generation of MMCP to determine the queue to which the message is to be linked during input queueing.

If the queue parameter is omitted, queue number 1 is assumed.

■ Positional parameter 3:

no-message-available-address

This label is the address of an optional user routine to which control is transferred if there is no message available in the input queue. If positional parameter 3 is omitted, control is not returned to the user until a message has been placed in the work area.

The work area length must be at least that of the buffers defined in the MMCP generation macro instruction (2.1.1). The work area will contain a 2-byte message length, an SOH character, the 3-byte symbolic terminal identifier as defined in the MMTRM macro instruction (2.1), followed by STX through ETX. Figure 2-1 gives an example of a message from a UNISCOPE 100.

BYTES	0	1	2	3
		14	SOH	T
BYTES	4	5	6	7
	1	S	STX	ESC
BYTES	8	9	10	11
	VT	Y	X	NUL
BYTES	12	13	14	15
	SI	SOE	S	T
BYTES	16	17	18	19
	A	R	T	ETX

LEGEND:

- Bytes 0-1 Message length in binary
- Byte 2 SOH character
- Bytes 3-4 Symbolic terminal identification
- Byte 5 UNISCOPE 100 screen identification
- Byte 6 Start of text character
- Bytes 7-12 Coordinate control characters
- Byte 13 Start of entry character
- Bytes 14-18 Message
- Byte 19 End-of-text character

Figure 2-1. Message Received from GETMM

If INTERCPT=YES has been specified in the MMCP generation macro instruction (2.1.1), the message presented in the user work area may be a message that the user has previously queued as output. A bit in the error half word will indicate this. In this case, the transmission of the message was not possible due to hardware failure (terminal, multiplexer, or line), and the message is being returned to the user for possible rerouting. If INTERCPT=NO was specified, these messages are discarded. The GETMM macro instruction produces the following expansion:

LABEL	Δ OPERATION Δ	OPERAND	COMMENTS
*LAB3	GETMM	WORK, 2, NOMSG	
	CNOP	2, 4	Adjust to boundary
LAB3	BAL	R1\$, *+16	
	DC	H'0'	Error half word
	DC	X'3'	GETMM request identifier
	DC	AL3(WORK)	Work area address
	DC	A(NOMSG)	No MSG transfer address
	DC	XL2'2'	Queue number
	SVC	SV\$UPR	

Specifying the operand (1) generates the SVC instruction.

If the GETMM macro instruction has not been completed successfully, bits will be set in the error half word to reflect the following hexadecimal configurations identifying the reasons for failure:

Reason	Half-word format
Request unknown	X'8000'
Second user attempting interface with MMCP	X'4000'
No line open	X'2000'
Message is from intercept queue	X'1000'
Message too long	X'0800'
Invalid queue number	X'0400'
MMCP not in main storage	X'0100'

2.2.3. Send Output Message Macro Instruction (PUTMM)

The PUTMM macro instruction is used to place a complete message on the output queue of a particular line. The user may send the same message to a number of terminals; these terminals are listed by the user. The formats of the PUTMM instruction are as follows:

LABEL	△ OPERATION △	OPERAND
[name]	PUTMM	{ work-area-name,terminal-list-name } (1)

where:

name

Is a user-defined label.

■ Positional parameter 1:

work-area-name

Is the label of a user-defined work area containing a 2-byte message length and the message to be transmitted consisting of STX through ETX. The maximum work area length is 4 bytes less than the user-defined buffer length in the MMCP macro instruction (2.1.1).

(1)

The user preloads register 1 with the address of a packet in the following format:

CNOP	2,4	Alignment
DC	H'0'	Error half word
DC	X'4'	Request identifier
DC	AL3 (work-area-name)	
DC	A(terminal-list-name)	

■ Positional parameter 2:

terminal-list-name

Is the label of a user-defined destination terminal list consisting of a half-word binary count of the number of destinations followed by a list of 4-byte entries. Each entry consists of a 3-byte symbolic terminal identifier plus a 1-byte status (set by the user to 0) which must be checked by the user if the error half word indicates that the packet has not been completely processed. The 3-byte terminal identifier consists of a 2-byte terminal name defined by MMTRM (2.1.4), and a 1-byte device identifier (Appendix A).

The following two examples indicate two possible user PUTMM macro instructions. The first sends the message in work area 'WORK' to four terminals, symbolic terminal identifiers TAS, T2S, T3S, and T4S. The second example sends the message in work area 'WORK' to terminal TAS.

■ Example 1:

LABEL	OPERATOR	COMMAND	
1			
HEADER	PUTMM	WORK,LISTA1	
LISTA1	DC	H'4'	NUMBER OF TERMINALS
	DC	C'TAS'	TERMINAL 1
	DC	X'0'	STATUS
	DC	C'T2S'	TERMINAL 2
	DC	X'0'	STATUS
	DC	C'T3S'	TERMINAL 3
	DC	X'0'	STATUS
	DC	C'T4S'	TERMINAL 4
	DC	X'0'	STATUS

■ Example 2:

LABEL	OPERATOR	COMMAND	
1			
QUERY	PUTMM	WORK,LISTA2	
LISTA2	DC	H'1'	NUMBER OF TERMINALS
	DC	C'TAS'	TERMINAL
	DC	X'0'	

Figure 2-2 gives an example of a user message to a UNISCOPE 100.

BYTES	0	1	2	3
		12	STX	ESC
BYTES	4	5	6	7
	VT	Y	X	SI
BYTES	8	9	10	11
	E	N	D	ETX

LEGEND:

- Bytes 0-1 Message length in binary
- Byte 2 Start of text character
- Bytes 3-7 Coordinate control character
- Bytes 8-10 Message
- Byte 11 End-of-text character

Figure 2-2. Message Sent by PUTMM

The PUTMM macro instruction produces the following expansion:

LABEL	Δ OPERATION Δ	OPERAND	COMMENTS
*LAB4	PUTMM	WORK,LIST	
	CNOP	2, 4	Adjust to word boundary
LAB4	BAL	R1\$,*+14	
	DC	H'0'	Error half word
	DC	X'4'	PUTMM request identifier
	DC	AL3(WORK)	Work area address
	DC	A(LIST)	Terminal list address
	SVC	SV\$UPR	

Specifying the operand (1) generates the SVC instruction.

If the PUTMM macro instruction has not been completed successfully, bits will be set in the error half word to reflect the following hexadecimal configurations identifying the reasons for failure:

<u>Reason</u>	<u>Half-word Format</u>
Request unknown	X'8000'
Second user attempting MMCP interface	X'4000'
Packet not completely processed	X'2000'
Message too long	X'1000'
Insufficient buffers for multidestinational message	X'0800'
No EOM	X'0400'
No STX	X'0200'
MMCP not in main storage	X'0100'
MSG too short (minimum length message must be 4 bytes)	X'0080'
All buffers assigned to input	X'0040'

If all the buffers are assigned to input, the user should execute a GETMM macro instruction to free buffers queued for input.

If the packet-not-completely-processed bit is set in the error half word, the terminal status byte of each terminal in the PUTMM list should be checked to determine which terminal's messages have been completely processed and which have not. The terminal status byte will indicate the reason for failure by specifying the following hexadecimal configurations:

<u>Reason</u>	<u>Byte Format</u>
Unknown terminal identifier	X'80'
Unknown device identifier	X'40'
Line not open	X'20'
Terminal down	X'10'
Line down	X'08'

2.2.4. Line Close Macro Instruction (CLSMM)

The CLSMM macro instruction is used to complete output of any messages already queued for the line and to discontinue polling. Upon return from the CLSMM macro instruction, it is the responsibility of the user program to check the error half word generated by the expansion of the macro instruction. The formats of the CLSMM macro instruction are as follows:

<u>LABEL</u>	<u>Δ OPERATION Δ</u>	<u>OPERAND</u>
[name]	CLSMM	{ linename } { (1) }

where:

name
Is a user-defined label.

- Positional parameter 1:

linename
Is a 3-character symbolic name of the relevant DTLNE line macro instruction.

(1)
The user preloads register 1 with the address of a packet in the following format:

DC	H'0'	Error half word
DC	X'2'	Request identifier
DC	CL3'linename'	

The line close macro instruction produces the following expansion:

LABEL	Δ OPERATION Δ	OPERAND	COMMENTS
*LAB7	CLSMM	LN1	
	CNOP	2, 4	
LAB7	BAL	R1\$, *+10	
	DC	H'0'	Error half word
	DC	X'2'	Close line indicator
	DC	CL3'LN1'	Linename
	SVC	SV\$UPR	

Specifying the operand (1) generates the SVC instruction.

If the line close function has not been completed successfully, bits will be set in the error half word to reflect the following hexadecimal configurations identifying the reason for failure:

<u>Reason</u>	<u>Half-word Format</u>
Request unknown	X'8000'
Second user attempting interface with MMCP	X'4000'
Invalid linename	X'2000'
Line not open	X'1000'
MMCP not in main storage	X'0100'



3. Supervisor Generation

The OS/4 communications supervisor is required to operate the mini message control program (MMCP). The communications supervisor must be generated from the latest procedure library.

At system generation time, a supervisor must be generated which incorporates the message control interfaces. When generating the supervisor, the following rules pertaining to the MMCP should be noted:

- The STDEQU macro instruction must contain the MMCP parameter.
- The SYSTEM macro instruction must contain the MMCP parameter.
- The following macro instruction must appear immediately after the SYSTEM macro instruction:

LABEL	Δ OPERATION Δ	OPERAND
	MCPGEN	IST=n

where:

n

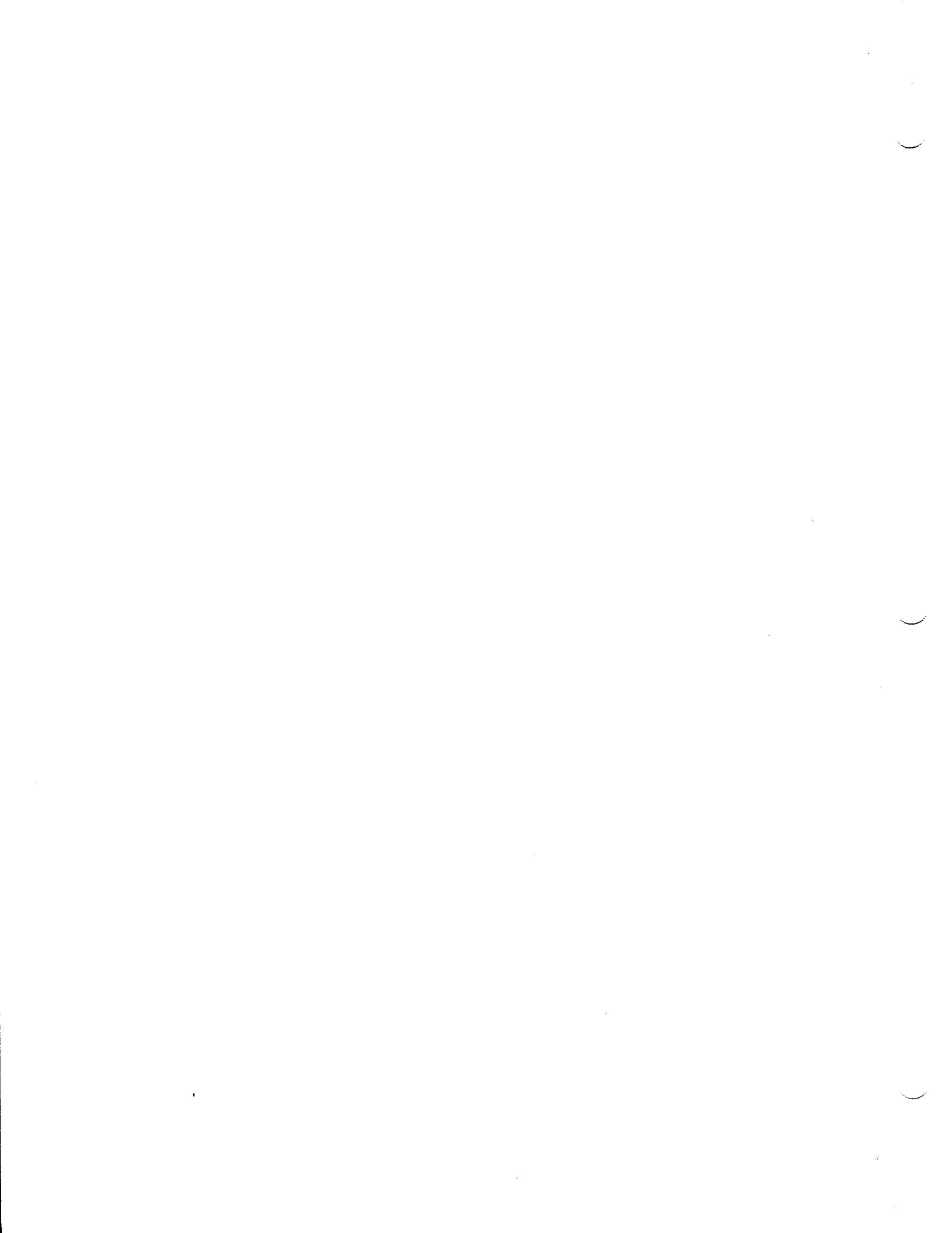
Is the number of communication lines that can be active at any one time. If n is missing or less than 1, n is set to 4. If n is greater than 64 (maximum), n is set to 64.

The following statement must appear in the control stream to prevent write-protect errors in the MMCP:

```
// OPTION SUPVMODE
```

The following statement must appear in the control stream to provide a system dump when a failure occurs:

```
// OPTION SYSDUMP
```



4. Operator Communications

All mini message control program (MMCP) operator message formats are preceded by an operating system printed prefix of @Δhh:mmΔ for unsolicited type-ins and lhh:mmΔjjΔ for MMCP initiated type-outs

where:

- hh**
Represents the hour.
- mm**
Represents minutes.
- jj**
Represents the job number.

4.1. UNSOLICITED TYPE-INS

The operator has the facility for making unsolicited type-ins to MMCP in order to set terminals up and down or to cancel MMCP. The operator message formats are as follows:

Format 1:

jjEOM,Δxx,yyy

Format 2:

jjEOM,ΔCANCEL

Format 1 message should be constructed as follows:

- jj**
Is the job number.
- EOM**
Is the end of message character.
- Δ**
Is the response from the operating system when the type-in can be accepted.

xx

Is the operation code as follows:

- LD Set line down
- LU Set line up
- TD Set terminal down
- TU Set terminal up

yyy

Is the 2-character symbolic terminal name or the 3-character linename.

Format 2 is used to cancel MMCP. The same explanation of jj, EOM, and ,Δ in format 1 also applies to format 2.

4.2. MMCP TYPE-OUTS

MMCP initiates type-outs, which inform the console operator of hardware malfunctions and MMCP status. MMCP initiated type-outs have the following format:

B\$nn message

where:

nn

Designates nine 2-digit type-out codes from B\$00 through B\$08. Each MMCP initiated type-out message is explained as follows:

B\$00 MINI-MCP VER nn.nn READY

Explanation:

This message is printed out when MMCP has been initialized. nn.nn represents the current version number of MMCP.

Action:

Proceed with operations.

B\$01

Explanation:

This is printed as a response to an invalid type-in caused by an invalid message type, an invalid terminal, or an attempt to put up (or down) a terminal or line which already has up or down status.

Action:

Check the type-in message format, the validity of terminal or linename, and the previous status of terminal or line (up or down).

B\$02 `linename,△error-flag-bytes`

Explanation:

This message indicates a line failure. The contents of the user error flag bytes (2.2.1.1) indicate the reason for failure and the subsequent action taken. This type-out specifies that the line has been set down.

Action:

If the hardware error is rectified, the line may be set up by an operator type-in.

B\$03 `terminal-name[,P],error-flag-bytes`

Explanation:

This message indicates a terminal failure. If printed, the character P indicates that the associated poll entry has been disabled and subsequent action taken. The contents of the user error flag bytes indicate the reason for failure. This type-out specifies that the terminal has been set down.

Action:

If the hardware error is rectified, the terminal may be set up by an operator type-in.

B\$04 **MINI-MCP CANCELLED**

Explanation:

This message is typed out subsequent to a cancelled type-in or a program check condition.

Action:

If a program check condition exists, the following bits in the BC\$REQ flag byte should be checked to determine the reason for cancellation:

Bit	Reason for Failure
3 = 1	Lost interrupts
4 = 1	Unidentified device address on interrupt

This byte can be found at X'2B2' beyond the start of MMCP.



B\$05 NO BUFFERS

Explanation:

This message occurs when MMCP has been generated without the keyword NUMBUF specified in the MMCP macro call and insufficient main storage has been allocated at run time to generate any buffers.

Action:

MMCP must be reloaded and allocated more main storage space.

The three remaining console messages generated by MMCP are for disc queueing:

B\$06 CYLS nnn Y/N

Explanation:

This message indicates that the number of cylinders requested for disc queueing is not available. The actual number of cylinders available is indicated in the message by nnn.

Action:

The user has the option to continue the jobs with the disc space available by replying Y(yes), or to cancel the job by replying N(no).

B\$07 SYSPPOOL ERR

Explanation:

This message indicates that SYSPPOOL has been incorrectly specified and MMCP automatically cancels.

Action:

Job control statements should be checked.

B\$08

Explanation:

This message indicates that an unrecoverable disc error has occurred and is for information only.

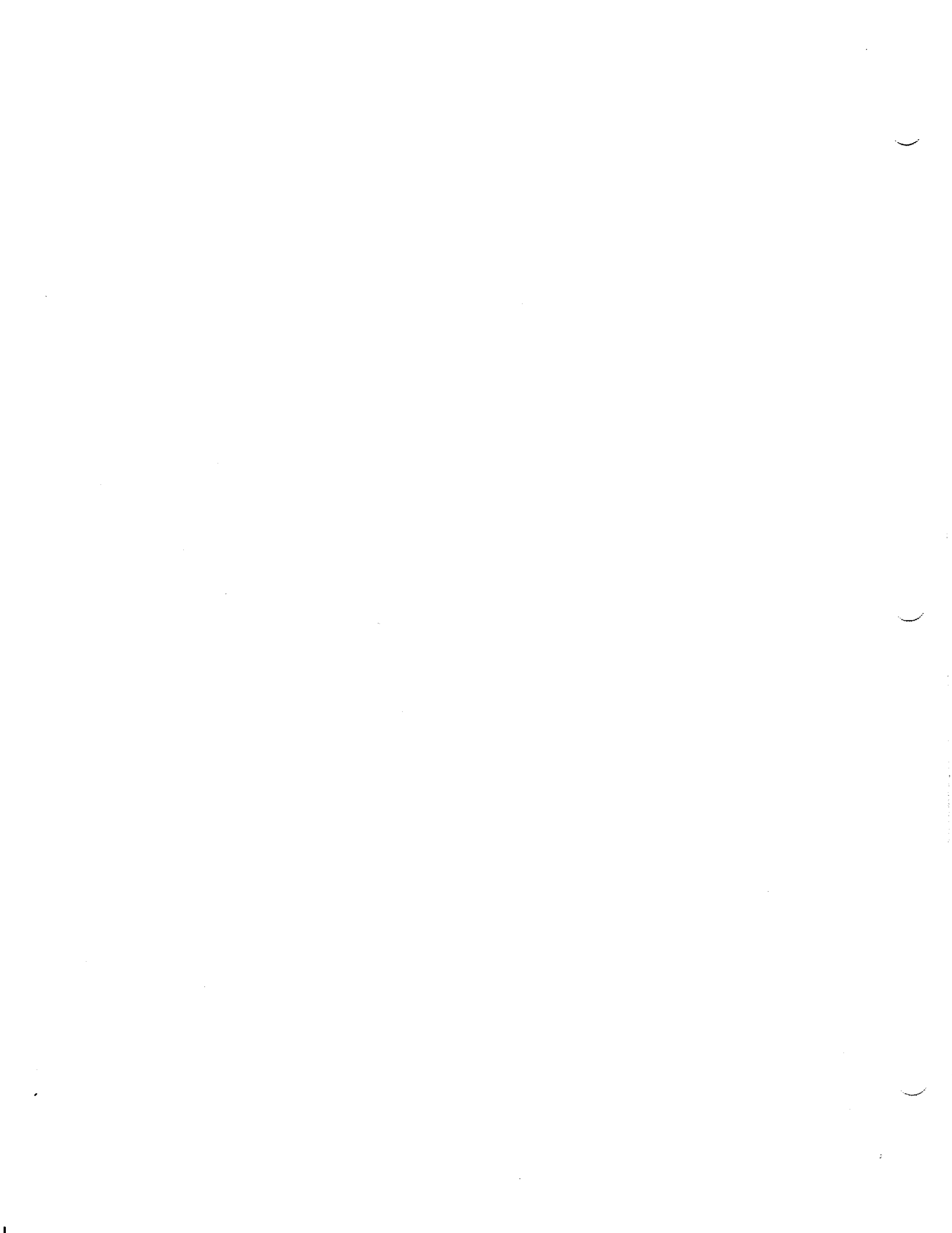
Action:

No action is required.

Appendix A. Symbolic Device Identifiers

The following are the symbolic device identifiers used in conjunction with a terminal name when sending a message:

- S UNISCOPE 100 Display Terminal screen
- P UNISCOPE 100 Display Terminal printer
- K UNIVAC DCT 1000 Data Communications Terminal keyboard
- L UNIVAC DCT 1000 Data Communications Terminal printer
- C UNIVAC DCT 1000 Data Communications Terminal card reader
- V UNIVAC DCT 1000 Data Communications Terminal VIP card punch
- R UNIVAC DCT 1000 Data Communications Terminal paper tape reader
- T UNIVAC DCT 1000 Data Communications Terminal paper tape punch



Comments concerning this manual may be made in the space provided below. Please fill in the requested information.

System: _____

Manual Title: _____

UP No: _____ Revision No: _____ Update: _____

Name of User: _____

Address of User: _____

Comments:

CUT

FOLD

FIRST CLASS
PERMIT NO. 21
BLUE BELL, PA.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

UNIVAC

P.O. BOX 500

BLUE BELL, PA. 19422

ATTN: SYSTEMS PUBLICATIONS DEPT.

CUT

FOLD