MRX/OS Telecommunications

Reference Manual

2200.007-01

MEMOREX

Computer System Products

May 1973 Edition

This edition is a major revision and obsoletes all previous editions. It documents the telecommunications programming system at its level in MRX/OS Release 2.

Technical changes to text, tables, and figures are marked with a vertical bar in the margin. Changes due to subsequent releases will be documented in future publications bulletins or revisions.

Requests for copies of Memorex publications should be made to your Memorex representative or to the Memorex branch office serving your locality.

A reader's comment form is provided at the back of this publication. If the form has been removed, comments may be addressed to the Memorex Corporation, Publications Dept., 8941 — 10th Ave. No. (Golden Valley) Minneapolis, Minnesota 55427.

PREFACE

This publication provides information necessary to program the MRX/40 or 50 System for telecommunications applications at the logical or physical level. The programmer using this manual is assumed to have experience in telecommunications programming. Chapter 11 contains information on Control Language statements for telecommunications. The expansions of the various macros and control blocks, and error recovery procedures are covered in the appendixes.

GLOSSARY

Asynchronous Data Transfer — In this transmission method, each byte is sent separately, with a preceding start bit and one or two following stop bits to provide hardware synchronization.

Duplex Line — A communications line that can transmit data in both directions at the same time.

Half-Duplex Line — A communications line that can transmit data in either direction, but not at the same time.

Integrated Communications Adapter — A communications hardware adapter that converts data on a communication line into a form that is acceptable to the processing unit, and vice versa. The functions of the ICA include assembly of bits into characters, checking data for validity, inserting and stripping of checking characters, and buffering of data. The ICA performs functions similar to a transmission control unit.

Local Line — A communications line that only covers a short distance (for the MRX/40 and 50 systems the maximum length of local lines is 50 ft.). Local lines connect directly to the ICA, do not require modems, and are used without dialing or operator intervention.

Logically Enabled — The hardware has been primed for connection, but connection has not been established.

Modem — A device leased or bought from a common carrier that converts the signals of a terminal to a signal that can be transmitted on a communications line, and vice versa. This terminal-to-line and line-to-terminal conversion is called MOdulation/DEModulation and is required to send signals over any distance. Modems may also be referred to as data sets.

Non-Switched Line — A communications line that connects distant points (requiring modems), and has a leased line connection between the terminal and ICA. These lines do not require dialing or operator intervention for message transfer.

Physically Enabled — A connection has been established.

Simplex Line — A communications line that can transmit data in one direction only.

Switched Line — A communications line that connects distant points (requiring modems) and requires dialing or operator intervention to connect the terminal and ICA.

Synchronous Data Transfer — In this transmission method, a series of bytes is sent at one time. There are no start-stop bits for each byte; instead, each block of bytes is preceded by synchronization characters.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The Memorex Telecommunications System is a combination of software and hardware that provides the user with the ability to communicate between central and remote locations.

There are two levels of programming in the telecommunications system, logical programming (Logical TCOM) and physical programming (Physical TCOM). The user communicates with both of these programs with two distinct sets of macros. The difference between these two programming levels is that Physical TCOM is at a much more basic level of programming than Logical TCOM.

The Physical TCOM user is concerned with such tasks as framing messages, translating messages, requesting I/O service, error recovery procedures, and writing command programs. Although the Physical TCOM user must be concerned with such basic functions, he has the advantage of being able to write programs for any terminal, even those not supported by Logical TCOM.

Logical TCOM provides all the basic functions for the programmer, so that he can be more concerned with direct communication with the remote terminal. If the Logical TCOM system is somewhat easier to use than Physical TCOM, it also occupies more storage than Physical TCOM. The programmer may use either or both of these programming levels depending on the nature of his applications program.

Data transfer in the telecommunications system may be by synchronous or asynchronous transmission, over switched or non-switched lines, using USASCII or EBCDIC code. For complete information on transmission conventions, refer to *Chapter 3. Transmission and Line Definitions.*

2. HARDWARE CONFIGURATION

MINIMUM CONFIGURATION

Telecommunications can be supported by any Memorex/40 or 50 system with a minimum of 24K bytes of storage. The 24K byte system supports only physical communications, one switched or non-switched line, and asynchronous and nontransparent synchronous data transmission.

MAXIMUM CONFIGURATION

Memory capacities larger than 24K bytes will permit full use of the MRX/40's seven communication lines or the MRX/50's fifteen communications lines. The lines can be connected to switched or non-switched lines, carrying asynchronous or nontransparent synchronous data.

SYSTEM STORAGE REQUIREMENTS

Appendix B contains the information needed to determine the Physical and Logical TCOM system storage capacities. (The storage required for individual macro expansions is given in the discussion of each macro in *Chapter 6. Logical Communications Macros* or *Chapter 8. Physical Communications Macros.)*

TERMINALS SUPPORTED BY TCOM

The asynchronous terminals in Table 2-1 are supported by TCOM.

Table 2-1. Asynchronous Terminals Supported by TCOM

Terminal	Baud	Code (ASCII)
MRX 1240 – half duplex	110/150/300/600	odd/even/no parity
MRX 1250	110/150/300/600	odd/even/no parity
MRX 1280	110/150/300/600/1200	odd/even/no parity
TTY 33 KSR and ASR	110	odd/even/no parity
TTY 35 KSR and ASR	110	even/no parity
TTY 37 KSR and ASR	150	odd/even/no parity
TTY 38 KSR and ASR	110	odd/even/no parity

Synchronous terminals which operate at speeds of 600 to 4800 baud are supported in a transparent or nontransparent EBCDIC mode or a nontransparent ASCII mode.

MODEM/LINE TYPES SUPPORTED BY TCOM

The modem/line types shown in Table 2-2 are supported by TCOM.

Table 2-2. Modem/Line Type Descriptions

Modem/ Line Type	Half/ Full Duplex	Async/ Sync	Transp/ Non-Transp	Modem	Leased/ Switched	Answer & Manual Dial/ Calling	Recommended Speeds
A6	H	S	NT	WE 201A WE 202C,D	S	A	201A – 2000 BPS 201B – 2400 BPS 202C, D – 600 or 1200 BPS 203A – 1800, 2400, 3600 or 4800 BPS 208A – 4800 BPS
A8	H	S	NT	WE 202C,D 2-Wire Secondary Channel	L	–	
AA	H	S	NT		S	A	
B4	H	S	T	WE 201A,B WE 202C,D WE 203A WE 208A	L	–	
B5	H	S	T	WE 201A,B WE 202C,D WE 203A WE 208A } Multipoint	L	–	
B6	H	S	T	WE 201A WE 202C,D	S	A	
B8	H	S	T	WE 202C,D 2-Wire Secondary Channel	L	–	
BA	H	S	T		S	A	
80	H	A	–	NONE (Local EIA-RS-232-C)	L	–	1240 – 600 or 1200 BPS
84*	H	A	–	WE 103A,E,F WE 113B WE 202C,D 4-Wire	L	–	

Table 2-2. Modem/Line Type Descriptions (Continued)

Modem/ Line Type	Half/ Full Duplex	Async/ Sync	Transp/ Non-Transp	Modem	Leased/ Switched	Answer& Manual/Dial/ Calling	Recommended Speed
85	H	A	–	WE 103F WE 202C,D 4-Wire } Multipoint	L	–	103A,E,F 113B 110,150 or 300 BPS
86	H	A	–	WE 103A,E WE 113B	S	A	202C,D – 600 or 1200 BPS
88	H	A	–	WE 202C,D	L	–	202C,D 600 or 1200 BPS
8A	H	A	–	2-Wire Secondary Channel	S	A	
8C	H	A	–	WE 202C,D	L	–	
8E	H	A	–	2-Wire	S	A	
98	H	A	–	Split Speed	L	–	600/75 or 1200/75 BPS
9A	H	A	–		S	A	
A4	H	S	NT	WE 201A,B WE 202C,D WE 203A WE 208A	L	–	201A – 2000 BPS 201B – 2400 BPS 202C,D – 600 or 1200 BPS 203A – 1800, 2400, 3600 or 4800 BPS 208A – 4800 BPS
A5	H	S	NT	WE 201A,B WE 202C,D WE 203A WE 208A } Multipoint	L	–	

*84 is the line type used to describe direct line connections which do not include modems at 10, 15, or 30.

3. TRANSMISSION AND LINE DEFINITIONS

LINE CONFIGURATION

Data may be transferred between a remote location and the computer over local, non-switched or switched lines.

Local lines cover a short distance, and thus do not require modems. They connect directly to the Integrated Communications Adapter (ICA), and do not require dialing or operator intervention for message transfer.

Non-switched lines connect distant points, requiring modems, and have a direct connection from the terminal to the ICA. These lines do not require dialing or operator intervention for message transfer.

Switched lines connect distant points, requiring modems, and require dialing or operator intervention to connect the terminal and the ICA. A switched connection may be made from the terminal or from the computer.

Figure 3-1 shows a block diagram of local, non-switched, and switched line connections.

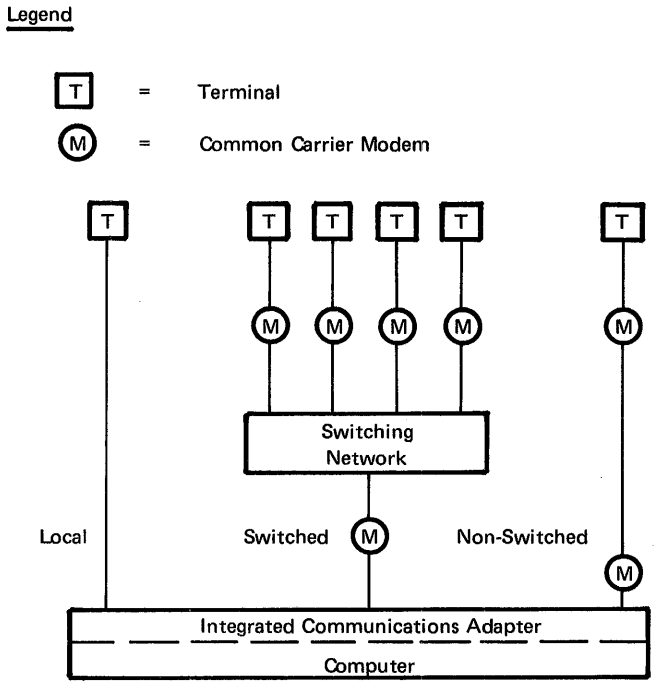
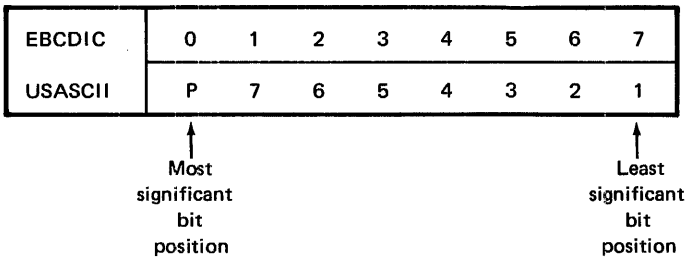


Figure 3-1. Line Configuration

TRANSMISSION CODES

MRX/OS Telecommunications (TCOM) supports the Extended Binary Coded Decimal Interchange Code (EBCDIC) and the United States of America Standard Code for Information Interchange (USASCII). When programming at the logical level, all received USASCII data is translated by the system into EBCDIC before being placed in the user's storage area. Likewise, all EBCDIC data in a user's storage area is translated by the system to USASCII before being placed on an asynchronous line. Under physical level programming, the user is responsible for code translation.

The following illustration shows the bit positions for the EBCDIC and USASCII codes.



The USASCII code may be transmitted with even, odd, or no parity. Only one parity convention per line is permissible at the logical level, but at the physical level the parity convention can be altered.

Under USASCII odd- or even-parity transmissions, the hardware adds the parity bit before the bits are transmitted to the line. (When transmitting without parity, no parity is checked.) USASCII data received by the computer is checked for good parity. If the parity is good, the parity bit is stripped from the byte before the byte is stored in the buffer. Bytes with faulty parity have a one-bit inserted in the parity position before the byte is stored in the buffer. Therefore, the physical level programmer must test the parity bit to determine bad parity.

Appendix A contains the hexadecimal assignments used for the EBCDIC and USASCII characters, printer graphics, and control symbols used during data transmission.

TRANSMISSION CONVENTIONS

Line data transfer occurs by asynchronous or synchronous transmission.

In asynchronous data transfer, each byte in the transmitted message is sent as an entity. One byte comprises one start bit followed by the data code and one or two stop bits. The start and stop bits are added by the hardware at transmission time and are used to synchronize the hardware at the receiving end. Figure 3-2 illustrates an asynchronous data byte with one stop bit. One asynchronous message will comprise many asynchronous bytes with the proper message framing text and line control characters. In asynchronous transmission, control characters normally terminate data transmission; however, the use of control

characters is governed by the terminal used and not by convention. The user may frame his own message if he chooses physical level programming, or he can select logical level programming, and specify the framing characters required to frame his messages.

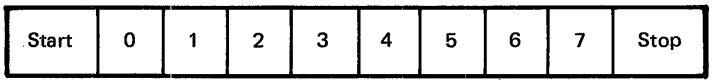


Figure 3-2. Asynchronous Data Byte

Synchronous data transfer transmits a block of data at transmission time. No start or stop pulses are transmitted with each byte, but synchronization bytes precede each transmission. Those sync bytes are added by the hardware and used by the receiving device to establish sync prior to reception of data. Once line sync is established, the message is transmitted with its proper control characters for message framing and padding. Not all messages will contain a pad character. Pad characters are used only by certain receiving devices. The transmitted block ends with a hardware inserted block check character. The hardware uses this character to determine transmission errors. Error action taken by the system depends on the programming options selected by the user.

Control characters must precede and terminate each synchronous transmitted message. Since the message heading or text information could contain bytes that appear as termination control characters, premature termination of a message could occur. To prevent early message termination, a Data Link Escape (DLE) control sequence is performed at the start and termination of a message. This control operation causes all bytes that are control characters in the message to appear transparent to the Integrated Communications Adapter's control sensing logic. Messages under control of a DLE data control sequence are often referred to as transparent messages. Messages that are not under control of a DLE sequence are often referred to as nontransparent messages. Figure 3-3 illustrates a data block for nontransparent and transparent messages. Transparent text is not supported by the first release of MRX/OS, but will be supported in a later release.

Structuring the message for synchronous transmission can be the responsibility of the user or system. If the user decides to use physical-level programming, he must provide all timing, framing, and message translation needed by the receiving device. If he decides to leave structuring to the system by selecting logical-level programming, he need not worry about timing, framing, or message translation.

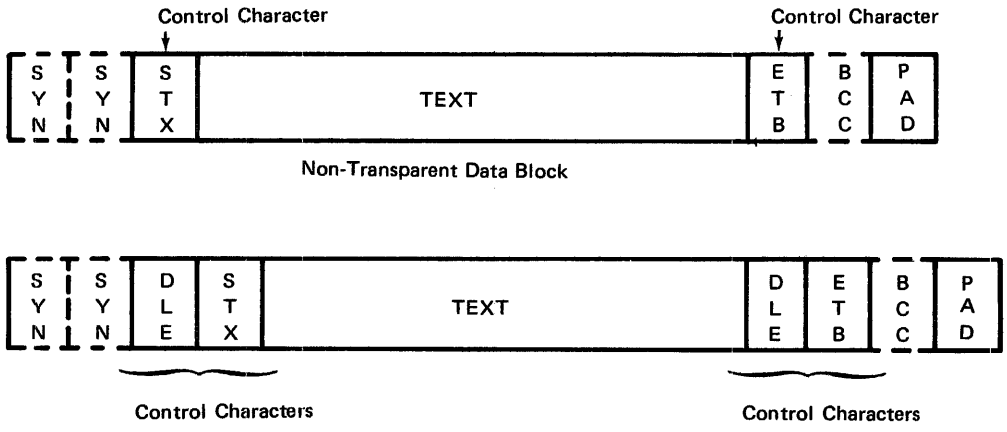


Figure 3-3. Synchronous Data Blocks — Transparent and Nontransparent

4. STANDARD SYMBOLS

The instructions in this manual are presented using certain symbols used to define which entries are required, required but named by the user, or optional. The chart below shows these symbols and defines their meanings.

ABCXYZ	Must appear in the coding as shown.
abcxyz	A required entry representing one name selected by the user.
{ NO / YES }	One and only one entry in the { } must be selected. An entry that is underlined is a default value, which is used if the programmer does not select any of the entries in the brackets. If no entry is underlined, there is no default value.
[LM=abc]	Items enclosed in [ ] are optional entries.
≋ ABCXYZ / abcxyz / L1279 ≋	One or more entries in ≋ ≋ must be selected. Limits on the selection of options are defined in the description of the macro using this notation.

NOTE

The brackets defined above are only used to present the instructions, they are not coded. Parentheses ( ) that appear in a statement are used to set off operand lists — these parentheses must be coded if the list format is used.

Figure 4-1. Standard Symbol Definition

5. LOGICAL COMMUNICATIONS PROGRAMMING

INTRODUCTION

The programmer communicates with the Logical TCOM program by using several macros to define the terminals and control message transfer. The physical-level control functions are all automatically provided by Logical TCOM (Logical TCOM uses Physical TCOM for these tasks), so that the user does not have to concern himself with line control, error recovery, and so forth. Logical TCOM provides the following services for the programmer: message framing, message translation, editing, and error recovery.

MESSAGE FRAMING AND RESPONSE CHARACTERS

Logical TCOM provides the STX, CR, LF, ETX, NUL, EOT, and ENQ output line control and message framing characters for asynchronous and synchronous transmission. However, during terminal communication, the user must provide all tabulation characters in his work area. Furthermore, if the asynchronous message contains more data than can be placed on a line, the user must provide control characters in his work area to return the carriage and advance paper before the line length is exceeded. For a detailed explanation of how Logical TCOM performs message framing for a terminal response, refer to Appendixes H and I.

MESSAGE TRANSLATION

Logical TCOM translates all received USASCII data to EBCDIC. Lower case received USASCII alphabetic characters are translated to upper case EBCDIC alphabetic characters. Transmitted messages that require translation from EBCDIC to USASCII are translated to support the terminal line code.

EDITING FUNCTIONS

All received USASCII coded messages are stripped of the STX, backspace, delete, rubout and null control characters, and edited in accordance with the control characters received. When a CAN or ENQ message delete code is sensed, the input message is prevented from passing to the user work area.

ERROR RECOVERY SERVICES

Logical TCOM performs all hardware error recovery and reports the success of each logical communications operation in the parameter packet's return code field. The user may also gain additional information about the operation by testing the Primary and Secondary status words in the terminal prefix area. It is the user's responsibility to terminate the line connection if an irrecoverable error is sensed and no user subroutine is available to assign the task to alternate terminals. (A detailed explanation of Logical TCOM error recovery can be found in Appendix J.)

LOGICAL TCOM RESTART

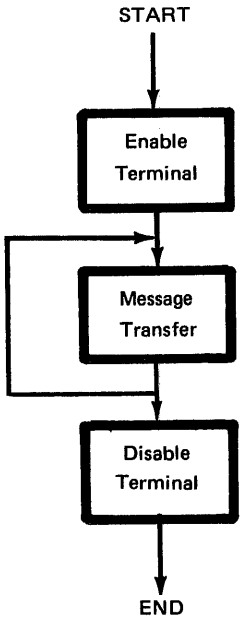
If a job step terminates abnormally, the operator is queried for permission to restart. If the reply is YES, lines and terminals will not be disabled and a restart is attempted from the last checkpoint. When the user program gains control after the restart, it must determine line and terminal status. Data in the buffer area defined by the Communications Master Control Block has not been reloaded or altered.

ITB FEATURE

Messages transmitted using the ITB feature are broken into individually transmitted segments of a specified length. Message segments received using ITB are reconstructed into whole messages with the ITB character edited out. See the TRMDEV macro ITBSIZ= keyword parameter.

MAIN FLOW OF A LOGICAL TELECOMMUNICATIONS SESSION

The flow of a logical communications session involves the user enabling the terminal lines by using the ENABLE macro, causing message transfer to any from the enabled terminal by using the SEND or RECEIVE macro, and disabling all enabled terminals by using the DISABLE macro. Figure 5-1 outlines this basic flow. Before a logical communications session begins, the user must define in his constant area one TERMINAL macro for each terminal or group of terminals used.



Each terminal used must be defined in the user's constant area (by a TERMINAL macro).

Figure 5-1. Main Flow of Logical Telecommunications Session

TYPES OF MACROS

Logical TCOM divides its macros into two categories. The first category consists of only the TERMINAL macro. The second category consists of the service request macros: ENABLE, DISABLE, SEND, and RECEIVE.

Each *TERMINAL* macro defined in the program constructs a user terminal control block (UTCB) which describes the environment within which the mentioned terminals will operate. This control block is used by the service request macros each time a logical communication operation is requested. An expansion of the UTCB and field description is located in Appendix E.

Each *service request macro* generates a service request instruction and a parameter packet. When the service request is complete, the Complete bit in the parameter packet is set and the success of the I/O operation is reported in the Return Code field. For a complete description of all the fields used in the parameter packet, the user should refer to the service request parameter packet expansion found in Appendix G.

DATA TRANSFER TECHNIQUES

The logical user using service request macros may continue processing after issuing the service request or delay processing until the service request is complete. If he chooses to continue processing after issuing the service request, he may not perform any processing on the affected terminal until the request is completed. The user can issue the WAIT macro to test for a completed service request, but once the WAIT macro is issued processing stops until the Complete bit is set.

SETTING OF THE COMPLETE BIT

All transmissions to or from a remote device pass through a buffer. When data is passed to or received from a buffer, the Complete bit is set in the parameter packet. The number of buffers and the size of each buffer for a remote device is defined by the MRX/OS Control Language //DEFINE statement.

IDENTIFYING THE SENDING TERMINAL

If more than one terminal is enabled, and the terminals are in a sending condition, issuing a RECEIVE macro causes the contents of the first filled buffer to be transferred into the user's work area. When this transfer occurs the parameter packet's Complete bit is set. To identify the sending terminal the user should examine the terminal name located in the user work area prefix. The work area prefix is always located immediately before the receiving work area. For an expansion of the work area prefix, refer to Appendix F.

IDENTIFYING THE RECEIVING TERMINAL

When sending information to a terminal, the user must insert the name of the receiving terminal in the work area prefix's terminal name field. If no message is to be transmitted,

but the user wishes to send a control sequence to the terminal that turns the line around, stops transmission, or returns the carriage and feeds paper, the user identifies the receiving terminal by placing the name of the receiving terminal in the UTCB prefix's name field. Insertion of the terminal name must occur before issuing the SEND macro.

ERROR REPORTING

After the execution of each service request macro, it is the responsibility of the user to test the validity of the communication request. The user evaluates the condition of the operation by testing the bits in the parameter packet's Return Code field. A detailed listing defining each bit designation in the Return Code field may be found in Appendix G. Each service request macro description contains a brief summary of the applicable return code bit designations.

The user may determine the line and terminal status by examining the Primary and Secondary Status words. The Primary Status field fills at the completion of a SEND or RECEIVE macro and, depending on the operand selected, is located either in the work area prefix field or in the UTCB prefix field. The location of the Primary Status field for various combinations of SEND and RECEIVE operands is listed with each macro description. The Secondary Status word is located in the UTCB and is filled when a receive status is issued. A complete description of the primary and secondary status data may be found in Appendix E with the UTCB expansion. The primary status bits are briefly described in the discussion of each service request macro; the secondary status bits are briefly described in the discussion of the RECEIVE macro (refer to *Chapter 6. Logical Communications Macros*).

LOCATING THE PARAMETER PACKET'S COMPLETE BIT AND RETURN CODE

The user may locate the parameter packet's Complete bit and Return Code by taking advantage of Logical TCOM's parameter packet labeling system. To use the label system the user assigns an alphanumeric label, beginning with an alphabetic character and being up to eight characters in length, to each service request macro designated. To all labels six characters or less, Logical TCOM appends two unique identifying characters. This process is repeated until all words except the sixth word of the parameter packet are labeled. Labels longer than six characters are truncated to six characters; therefore, the first six characters of a label should be unique. The truncated label is then appended as if it were a label of six characters or less. The appended characters are: SR for word one, FL for word two, CR for word three, OP for word four, CB for word five, and if message transfer occurs, WA for word seven. See Figure 5-2 for an example of this labeling system.

The user labels a service request macro TRM3RCV. To identify the parameter packet word containing the Complete bit and Return Code field, the user addresses the TRM3RCCR storage location.

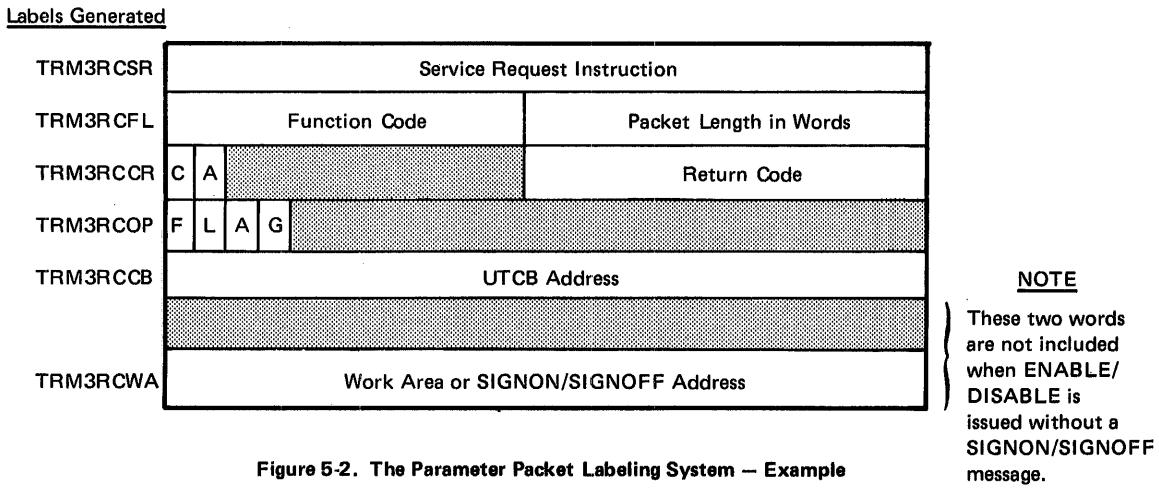


Figure 5-2. The Parameter Packet Labeling System — Example

If the user does not assign a name to a service request macro, Logical TCOM will assign unique labels for its own use. For complete information refer to *Service Request Name Fields* in *Chapter 6. Logical Communications Macros.*

COMPUTER TO TERMINAL DIALING PROCEDURES

To make a switched connection that is initiated by the computer, the user must provide a routine that informs the operator of the remote user's phone number and directs the operator to dial the number. This routine must be processed after the ENABLE macro is issued. The ENABLE macro completes when the remote terminal answers the call. The user must provide an alternate processing path if the remote terminal is unable to answer the call.

TRANSMISSION MODES

Three transmission modes: Conversational, Send, and Receive are possible for each terminal. If a terminal is in the Conversational mode, it may not be in the Send or Receive mode; however, a terminal not in the Conversational mode may be in the Send and/or Receive modes. Transmission mode is established by the CONVERS, SEND, and RECEIVE parameters in the TERMINAL macro.

In the *Conversational mode,* the user establishes a dialogue between the computer and the remote terminal with the terminal initiating the dialogue. After a line termination character is received, the device sending is required to receive.

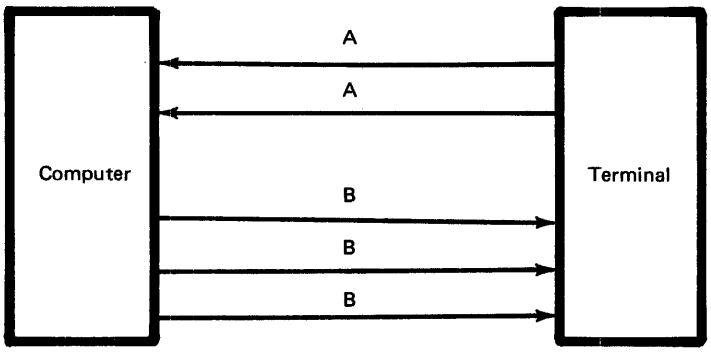
Users who wish only to send transmissions to a remote device establish a *Send mode* that ignores transmissions from the remote terminal. Likewise, users who establish a *Receive mode* are not permitted to send transmissions. If a user wishes to send and receive on a line but does not want to be bound by the transmission conventions established by the Conversational mode, he may use both the SEND and RECEIVE parameters in the TERMINAL macro, and initiation begins on the user's issuance of the SEND or RECEIVE macro.

TRANSMISSION SEGMENTS

Many times it is impossible to transmit a message with one transmission. This limitation is caused either by the number of printable characters that can be placed on a line, the number of characters that can be placed on a card, or the number of characters that can be transmitted to or from a terminal. Because of these limitations, messages often must be divided into segments and each segment is considered a complete transmission. Since each segment is considered a complete transmission, the user is given the option of changing the transmission direction at the end of any segment. Or, the user may transmit all the segments and change direction at the end of the message. If the user chooses to change direction at the end of a segment, he selects the TRANSUNIT parameter in the TERMINAL macro. If he chooses to change direction at the end of the message, he selects the MESSAGE parameter in the TERMINAL macro. The MESSAGE and TRANSUNIT parameters are entries for the CONVERS, SEND, and RECEIVE operands of the TERMINAL macro.

(All transmission segments, except the final segment, are terminated by the termination character designated by the TRMDEV macro's EOU or SEOU operands. Transmissions that are complete messages, or the final segment of a message, must be terminated by the termination character designated by the TRMDEV macro's SEOM and ENDLIST operands. A description of the TRMDEV macro can be found in *Chapter 9. The Terminal Catalog Routine.)*

Figure 5-3 shows how two messages, one from the terminal and one from the computer, would be handled if the TERMINAL macro's CONVERS = MESSAGE operand were selected. Data transfer in this example is always initiated by the terminal. The message from the terminal is segmented into two complete transmissions. At the end of the second transmission when the computer senses an EOM termination character defined by the TRMDEV macro's ENDLIST operand, line turnaround occurs and the computer responds with three complete transmissions followed by a message-terminating SEOM character.

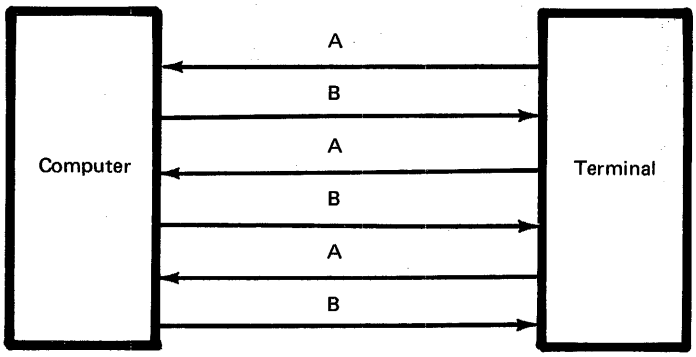


CONVERS=MESSAGE

Line turnaround can only occur after a complete message is sent.

Figure 5-3. CONVERS=MESSAGE Message Transfer

Figure 5-4 shows how two messages, each divided into three segments, are transmitted if the user selects the TERMINAL macro parameter, CONVERS=TRANSUNIT. Note, in this case, that line turnaround occurs after each segment of a message is sent or received.

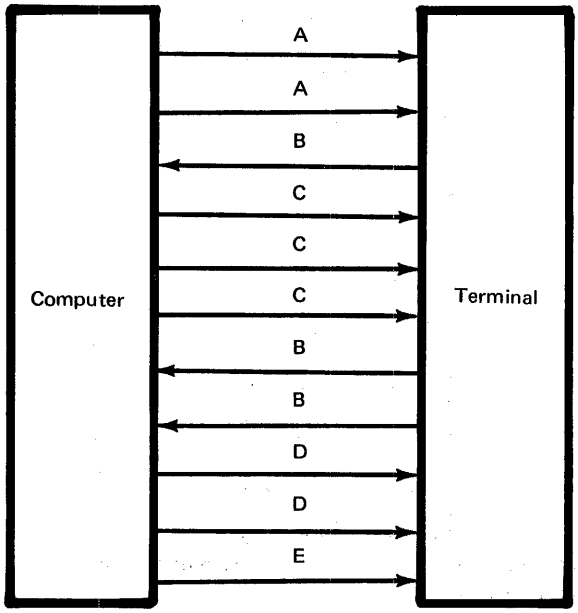


CONVERS=TRANSUNIT

Line turnaround must occur after each segment is sent.

Figure 5-4. CONVERS=TRANSUNIT Message Transfer

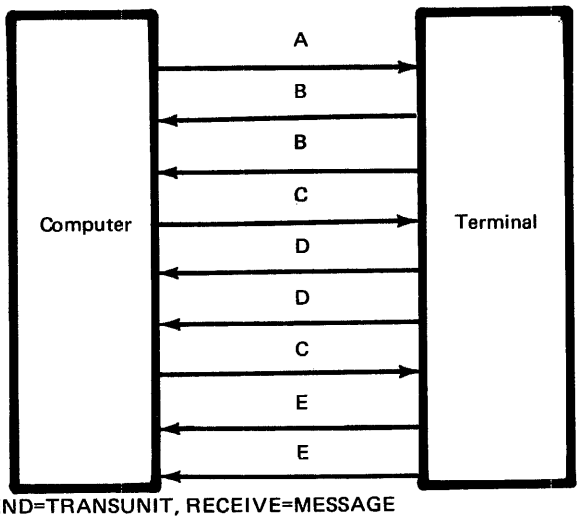
Figures 5-5 through 5-7 pictorially describe how message transfer takes place for various combinations of the SEND and RECEIVE operands using the MESSAGE or TRANSUNIT parameters in the TERMINAL macro. When in the Send and Receive mode, direction of the first transmission is determined by the user.



SEND=MESSAGE, RECEIVE=TRANSUNIT

Line turnaround can occur only after a complete message is sent. Messages received can have line turnaround after any segment (as for message B).

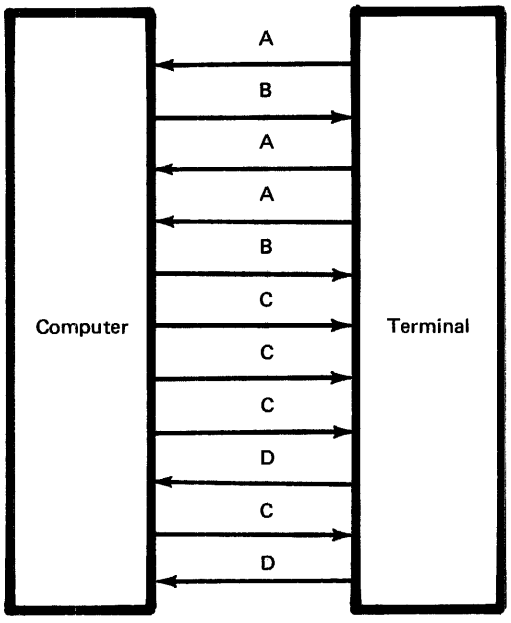
Figure 5-5. SEND=MESSAGE, RECEIVE=TRANSUNIT Message Transfer



SEND=TRANSUNIT, RECEIVE=MESSAGE

Line turnaround can occur only after a complete message is received. Messages sent can have line turnaround after any segment (as for message C).

Figure 5-6. SEND=TRANSUNIT, RECEIVE=MESSAGE Message Transfer



SEND=TRANSUNIT, RECEIVE=TRANSUNIT

Line turnaround can occur after any segment of any message sent or received.

Figure 5-7. SEND=TRANSUNIT, RECEIVE=TRANSUNIT Message Transfer

SYNCHRONOUS LINE INITIATION AND BIDDING PROCEDURES

Under synchronous operation, it is possible for both the remote terminal and computer to simultaneously initiate data transmission or bid for the line. The Logical TCOM TERMINAL macro defines which device will initiate transmission or gain control of the line when these simultaneous conditions occur.

Either the computer or the terminal can be designated as the initiator, and thus will initiate the transmission. If the computer is the initiator, the terminal is the non-initiator; if the terminal is the initiator the computer is the non-initiator. After line connection, the initiator transmits the first message.

In a similar manner, the computer or terminal may be named as master or slave. If the computer is designated as the master, and simultaneous line bidding occurs, the computer gains line control and the terminal is considered the slave. If the terminal is designated as master, the computer is considered the slave. When line contention occurs, the slave always yields to the master.

TERMINAL CATALOGING PROCEDURES

All terminals must be cataloged by the Logical TCOM user in a system library. For complete information, refer to *Chapter 9. The Terminal Catalog Routine.*

6. LOGICAL COMMUNICATIONS MACROS

INTRODUCTION

Logical level macros are divided into two groups. These groups deal with terminal definition and service request macros. Before using the material in this section, the user should be familiar with the information in *Chapter 5. Logical Communications Programming.*

TERMINAL DEFINITION MACRO

The TERMINAL macro constructs a user terminal control block (UTCB). This control block identifies the communicating terminals, and establishes the transmission mode. The information defined by the TERMINAL macro is used by the service request macros each time a logical communications operation is performed on a terminal. The expansion of the TERMINAL macro (the UTCB) is located in Appendix E.

TERMINAL – IDENTIFY TERMINALS

Name	Operation	Operands

label	TERMINAL	TERMNAM={terminal-name / (terminal-name,. . .terminal-name)}
		[,PREFIX=prefix-name]
		}[,CONVERS={MESSAGE / TRANSUNIT}]{
		}[,RECEIVE={MESSAGE / TRANSUNIT}]{
		}[,SEND={MESSAGE / TRANSUNIT}]{
		[,ATTENDD={YES / NO}]
		[,PROMPT={YES / NO}]
		[,CONTROL={MI / MN}{SI / SN}]

NAME

A required entry specifying the symbolic name of the TERMINAL operator used by the ENABLE, DISABLE, SEND or RECEIVE macros.

OPERANDS

TERMNAM= A required entry specifying the logical name of a terminal defined in a TRMDEV macro. If more than one terminal is specified, they must appear in parentheses separated by commas. Asynchronous and synchronous terminals may be grouped together; however, synchronous terminals must be grouped as master initiator, master non-initiator, slave initiator, or slave non-initiator terminals.

PREFIX= An optional entry tagging the beginning storage address of the UTCB's prefix status field. The data-name defining the storage address must start with an alphabetic character and be no longer than eight alphanumeric characters.

One or more of the operands CONVERS, RECEIVE, or SEND must be selected. However, selection of the CONVERS operand prohibits the use of the other operands.

CONVERS= This operand requires the user to send a response after receiving each message or transunit. The possible entries for this operand are MESSAGE or TRANSUNIT. The MESSAGE entry indicates that the entire message (all segments) must be sent before line turnaround can occur. The TRANSUNIT entry indicates that line turnaround can occur after any segment is sent. For a detailed discussion of messages and transunits, refer to *Chapter 5. Logical Communications Programming.*

(Termination characters for end of unit and end of message are defined by the user in the TRMDEV macro ENDLIST, EOU, SEOU, and SEOM operands. The TRMDEV macro is discussed in *Chapter 9. Terminal Catalog Routine.)*

RECEIVE= The user will receive one or more messages or transunits. The entries are MESSAGE or TRANSUNIT, as defined in the preceding text.

SEND= The user will send one or more messages or transunits. The entries are MESSAGE or TRANSUNIT, as defined in the preceding text.

ATTENDD= An optional entry for asynchronous terminals only. If NO is specified, the terminal is unattended, and it is assumed that all input is coming from the auxiliary device.

PROMPT= An optional entry. When omitted, PROMPT=YES is assumed and all asynchronous terminal receive operations are preceded by a transmitted control character sequence. This sequence is derived from the FFRAME, NL, NULLS, and SEOM operands of the TRMDEV macro and should only be used when receiving from an asynchronous terminal. This operand allows the user to advance the terminal's paper and print an * on the new line before receiving from the terminal.

CONTROL= An optional entry specifying whether the applications program controlling the synchronous terminal listed in the TERMNAM operand will be considered the master, slave, initiator, or non-initiator. An MI entry defines the applications program as the master initiator, MN as the master non-initiator, SI as the slave initiator, and SN as the slave non-initiator. When this operand is omitted, Logical TCOM defaults to MI. If only asynchronous terminals are listed in the TERMNAM operand, Logical TCOM ignores this operand.

EXPANSION STORAGE REQUIREMENTS

The number of words required by the TERMINAL macro can be calculated from the following formula.

Sw = 9+4t Where: Sw = Word storage requirements

t = Number of terminal names specified in the TERMNAM operand

SERVICE REQUEST MACRO INSTRUCTIONS

The ENABLE, DISABLE, RECEIVE, and SEND macro instructions provide an interface between the applications program and the Logical TCOM system. These macros allow the user program to request service that will enable and disable a line, transfer messages, and report terminal status. Each service request macro generates a service request instruction and a parameter packet. It is the service request instruction that requests communications service. The parameter packet consists of a list of parameters that further describe the request. The expansion for the service request macros is located in Appendix G.

SERVICE REQUEST NAME FIELD RESTRICTIONS

The optional name entry on all service request macros performs two functions: it labels the address of the first instruction in the macro expansion, and uses the first six characters (or less if the programmer used fewer than six characters) to provide unique names for the first, second, third, fourth, fifth, and seventh words of the service request parameter packet. The name field entry must begin with an alphabetic character and be no longer than eight alphanumeric characters; however, the first six characters must be unique for each service request macro.

If the user does not specify an entry in the name field the DISABLE, ENABLE, SEND, or RECEIVE macros, a label will be generated by Logical TCOM for the fields in the parameter packet.

The labels will be generated as follows:

XXYYYZZ Where: XX = $E for ENABLE
= $D for DISABLE
= $S for SEND
= $R for RECEIVE

YYYY = A unique 4 digit number from 0001 to 9999

ZZ = CR, FL, OP, CB, or WA depending on the field to be labeled

ENABLE – ENABLE TERMINAL

The first ENABLE activates the logical communications initialization routines; any logical communications macros issued prior to the first ENABLE will be returned with an error code.

The ENABLE macro also enables* one or more terminals listed in the referenced TERMINAL macro. The number of terminals enabled depends on the macro operands selected and the terminal line configuration. If each terminal listed in the referenced TERMINAL macro is assigned to a specific line, ENABLE connects all terminals. If more than one listed terminal is assigned to a switched line, the first terminal that is manually dialed or dials in is connected to the line. Only one terminal listed in the referenced TERMINAL macro is enabled if the PREFIX operand is selected. Once a terminal is enabled, it cannot be enabled again until it is first disabled. If a list of terminals, representing n lines, is enabled, all n lines are primed. After a terminal connects and then disconnects, the terminals that were previously candidates for connection are still candidates for connection (except the one that disconnected).

Name	Operation	Operands
[label]	ENABLE	TERMNAL={@R_n / name of TERMINAL macro} [,PREFIX={NO / YES}] [,SIGNON=({data-name / @R_n} {integer / @R_n})] [,RETURN={NO / YES}] [,LIST={NO / YES / null}] [,LISTNAM={@R_n / data-name}]

NAME

A required entry if LIST=YES; otherwise, an optional entry. For label restrictions, refer to *Locating the Parameter Packet's Complete Bit and Return Codes* in Chapter 5.

OPERANDS

TERMNAL= A required entry specifying the name of the TERMINAL macro defining the user terminal control block for the terminal(s) to be enabled. A general register can be used to address the defining TERMINAL macro if the address is loaded into the general register before the ENABLE macro is issued. The name of the TERMINAL macro must begin with an alphabetic character and be no longer than eight alphanumeric characters. If a register number is specified, the number must be preceded by the symbol @. Only one TERMINAL macro can be specified by the TERMNAL operand.

*ENABLE is not synonymous with connection. Dedicated lines are always connected – they must only be initialized for transmission. Switched lines must be primed to allow a terminal to dial in.

PREFIX= An optional entry. When coded PREFIX=YES, this operand notifies Logical TCOM of the one terminal listed in the referenced TERMINAL macro that is to be enabled. The name of the terminal to be enabled must be placed in the UTCB's current terminal name field before issuing the ENABLE macro. When this operand is omitted, PREFIX=NO is assumed and one or more terminals listed in the TERMINAL macro will be enabled. If the PREFIX keyword is specified for an ENABLE with LIST=YES, it must be specified for the complement ENABLE where LIST=NO.

SIGNON= An optional operand used when a sign-on message is to be transmitted to the terminals just enabled. The data-name defines the sign-on message's storage address, and the integer value represents the length of the message. The storage address must start with an alphabetic character and be no longer than eight alphanumeric characters. Sign-on message length is limited to the size of the available buffer defined by the Control Language //DEFINE statement BUFFER operand. General registers must be loaded with the storage address and message length before the macro is issued and used in place of the data-name and integer operands. If a register number is specified, the number must be preceded by the symbol @. The data in the message must be coded in line code as the sign-on message is not translated, edited, or framed.

RETURN= An optional entry that allows the user to continue processing after issuing an ENABLE. When this operand is omitted, RETURN=NO is assumed and the user's program will not receive control and must wait until the Complete bit in the parameter packet is set.

LIST= An optional entry that specifies whether the macro expansion is to generate the service request and/or the parameter packet.

If LIST=YES, only the parameter packet is generated. Other ENABLE operand requirements are as follows:

[TERMNAL=name of TERMINAL macro]
[,PREFIX={NO / YES}]
[,SIGNON={dataname,integer / YES}]

If there is a SIGNON message, either the SIGNON=YES or the SIGNON=dataname, integer operand must be used with LIST=YES because the length of the parameter packet generated depends on whether or not a SIGNON is present. SIGNON=YES reserves the space needed for the SIGNON parameter to be supplied later. SIGNON=dataname, integer actually supplies the values.

If LIST=NO, only the service request is generated. Any of the other ENABLE operands can be used with LIST=NO.

If LIST=null (no entry), both the service request and the parameter packet are generated. The default value for LIST is null.

LISTNAM= A required entry if LIST=NO is used. This operand specifies the address of the parameter packet associated with this service request.

TERMINATION CONDITIONS

When a list of terminals is being processed, any return code other than 00_{16} means that there was an error on at least one terminal. The return code describes the error condition for the last terminal processed in error.

Return Codes	Description
00	Normal completion (one or more terminals are enabled).
D0	Terminal(s)/line not defined by Control Language.
D1	No output buffer available for sign-on message.
D2	No input buffer of sufficient size is available.
D4	Macro usage error.
D5	Exception condition.
D8	Not enough space available at ENABLE time.
D9	The terminal name asked for with //DEF ID=xxxxxx, was not named in the TRMDEV macro which built the TDT descriptions on disc.
DA	Incorrect //DEF card in CLS.
DB	A terminal with an invalid ID attempted to connect.
DC	Request returned early because job step in terminating sequence.
DD	Irrecoverable Block I/O disc read error when reading SYSIN, //DEF tables, or the TDT descriptions from NUCLIB.

The secondary status data can only be received by issuing a status request after the ENABLE operation sets its Complete bit. The Primary Status word fills after the service request. If the ENABLE macro enables more than one terminal, the user will have to request status on each terminal to test the Primary and Secondary Status words.

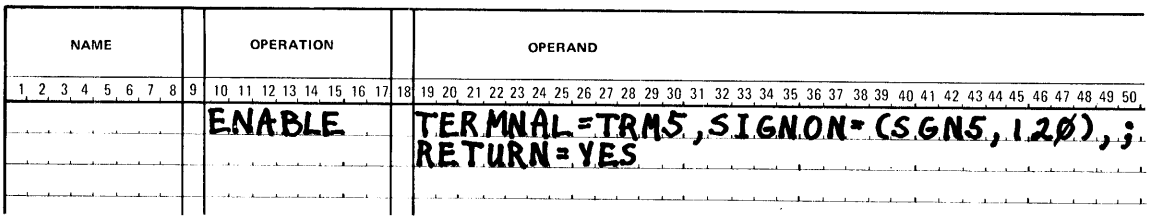
EXPANSION STORAGE REQUIREMENTS

To calculate the storage requirements, total the applicable storage quantities from the chart below.

Condition	Storage Required
The SIGNON operand and register notation are omitted (basic request).	5 words
The SIGNON operand is used.	2 words
The data-name is addressed by register notation.	2 words
The integer is addressed by register notation.	2 words
The integer keyword is used.	4 words
The TERMINAL macro is addressed by register notation.	2 words

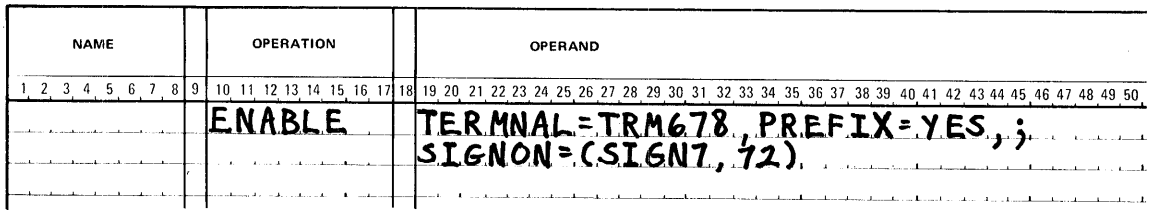
EXAMPLE ONE – ENABLE

The terminal to be enabled is specified in the TERMINAL macro named TRM5. A 120-character sign-on message will be used and is stored at location SGN5. Control returns to the program after the ENABLE operation is issued and before the logical enable operation completes.



EXAMPLE TWO – ENABLE

Three terminals called TRM6, TRM7, and TRM8 are specified in the TRM678 TERMINAL macro. Only TRM7 is to be enabled. The sign-on message is located in storage location SIGN7 and is 72 characters in length. The user chooses to wait until the enable is completed before continuing processing.



DISABLE – DISABLE TERMINAL

DISABLE breaks the line connection to one terminal or all terminals previously enabled and listed in the referenced TERMINAL macro. Any transmission in process or pending at the time the DISABLE is requested will continue to completion. The user may omit this macro and allow the system to disable line connections when terminating the job step; however, transmissions in progress will be aborted.

Name	Operation	Operands
[label]	DISABLE	TERMNAL={@Rn | name of TERMINAL macro} [,PREFIX={NO | YES}] [,SIGNOFF=({data-name | @Rn},{integer | @Rn})] [,RELEASE=YES] [,RETURN={NO | YES}] [,LIST={NO | YES | null}] [,LISTNAM={@Rn | data-name}] [,TAPE={NO | YES}]

NAME

A required entry if LIST=YES is specified; otherwise, an optional entry. For label restrictions refer to *Locating the Parameter Packet's Complete Bit and Return Code* in Chapter 5.

OPERANDS

TERMNAL= A required entry specifying the name of the TERMINAL macro defining the control block listing the terminals to be disabled. A general register can be used to address the defining TERMINAL macro if the address is loaded into the general register before the DISABLE macro is issued. The name of the TERMINAL macro must begin with an alphabetic character and be no longer than eight alphanumeric characters. If a register number is specified, the symbol @ must precede the register number.

PREFIX= An optional entry that notifies Logical TCOM that only one enabled terminal listed in the referenced TERMINAL macro is to be disabled. The name of the terminal to be disabled must be placed in the UTCB's terminal name field before issuing the DISABLE macro. When this operand is omitted, PREFIX=NO is assumed and all enabled terminals listed in the referenced TERMINAL macro are to be disabled. If the PREFIX keyword is specified for a DISABLE with LIST=YES, it must be specified for the complement DISABLE where LIST=NO.

SIGNOFF= An optional operand used when a sign-off message is to be transmitted to the terminals just before they are disabled. The data-name entry defines the storage address of the sign-off message and the integer entry represents the length of the message. The storage address must start with an alphabetic character and be no longer than eight alphanumeric characters. Sign-off message length is limited to the size of the available buffer defined by the Control Language //DEFINE statement BUFFER keyword. General registers may be loaded with the storage address and used in place of the data-name and integer operands. If a register number is specified, the symbol @ must precede the register number. The sign-off message data is translated to the proper line code by logical communications.

RELEASE= An optional entry that releases terminals from the applications programs making the released terminal available to other partitions.

RETURN= An optional entry that allows the user to continue processing after issuing a DISABLE. When this operand is omitted RETURN=YES is assumed and the user's program must wait until the Complete bit in the DISABLE macro's parameter packet sets before processing continues.

LIST= An optional entry that specifies whether the macro expansion is to generate the service request and/or the parameter packet.

If LIST=YES, only the parameter packet is generated. Any of the other DISABLE operands except RETURN and LISTNAM can be used with LIST=YES. If there is a SIGNON message, either the SIGNON=YES or the SIGNON=dataname, integer operand must be used with LIST=YES because the length of the parameter packet generated depends on whether or not a SIGNON is present. SIGNON=YES reserves the space needed for the SIGNON parameter to be supplied later. SIGNON=dataname, integer actually supplies the values.

If LIST=NO, only the service request is generated. Any of the other DISABLE operands can be used.

If LIST=null (no entry), both the service request and the parameter packet are generated. The default value for LIST is null.

LISTNAM= A required entry if LIST=NO is specified. This operand specifies the address of the parameter packet associated with this service request.

TERMINATION CONDITIONS

Return Codes	Description
00	Normal completion.
D0	Terminal(s)/line not defined by Control Language.
D1	No buffer available for sign-off message.
D5	Exception condition.
DC	Request not honored because job is in termination.

EXPANSION STORAGE REQUIREMENTS

To calculate the storage requirements, total the applicable storage quantities from the chart below.

Condition	Storage Required
The SIGNOFF operand and register notation are omitted (basic request).	5 words
The SIGNOFF operand is used.	2 words
The date-name is addressed by register notation.	2 words
The integer is addressed by register notation.	2 words
The integer keyword is used.	4 words
The TERMINAL macro is addressed by register notation.	2 words

EXAMPLE ONE – DISABLE

The terminal to be disabled is listed in the TRM5 TERMINAL macro. A 72-byte sign-off message is stored at location SGNF. The terminal is released from the applications program and control returns to the application program after the DISABLE is issued.

```
NAME      OPERATION   OPERAND
          DISABLE     TERMNAL=TRM5,SIGNOFF=(SGNF,72),;
                      RELEASE=YES,RETURN=YES
```

EXAMPLE TWO – DISABLE

One terminal, TRM5, listed in the TRM359 TERMINAL macro is to be disabled. Before issuing the DISABLE macro, terminal name TRM5 is loaded into the UTCB's terminal name field. There is no sign-off message. Terminal 5 will not be released from the applications program, and the application program will not continue until the DISABLE is completed.

```
NAME      OPERATION   OPERAND
          DISABLE     TERMNAL=TRM359,PREFIX=YES
```

RECEIVE – BUFFER TO WORK AREA

The RECEIVE macro transfers the next available message from the memory buffer to the user's work area. (The number and size of the memory buffers are defined by the Control Language //DEFINE statement BUFFER keyword.) If the STATUS operand is used, message transfer is inhibited; however, status information will appear in the designated terminal's UTCB prefix field.

Logical TCOM scans the list of terminals, returning to the beginning of the list when the end of the list is reached, until data is available from a terminal or until an error is detected on a terminal. Data is then moved into the memory buffer or an error code is posted. Scanning resumes with the next terminal when another RECEIVE is requested. When RECEIVE for a list is issued and a terminal is discovered in the scan whose next expected command is a SEND, the terminal will be skipped in the scan and no error will be reported. In the case of a single terminal operation, an error will be reported.

Name	Operation	Operands
[label]	RECEIVE	TERMNAL={@R$_n$ / name of TERMINAL macro} [,STATUS={NO / YES}] [,WKAREA={@R$_n$ / data-name}] [,RETURN={NO / YES}] [,PREFIX={NO / YES}] [,LIST={NO / YES / null}] [,LISTNAM={@R$_n$ / data-name}]

NAME

A required entry if LIST=YES; otherwise, an optional entry. For label restrictions, refer to *Locating the Parameter Packet's Complete Bit and Return Code* in Chapter 5.

OPERANDS

TERMNAL= A required entry specifying the name of the TERMINAL macro defining the UTCB. A general register can be used to address the defining TERMINAL macro if the address is loaded into the general register before the RECEIVE macro is issued. The name of the TERMINAL macro must begin with an alphabetic character and be no longer than eight alphanumeric characters. If a register number is specified, the symbol @ must precede the register number.

STATUS= An optional entry. STATUS=YES requests the current operational status of a terminal to appear in the UTCB's Primary and Secondary Status word fields. To request terminal status, the user must enter the name of the terminal that is to report status in the UTCB's terminal name field before issuing the RECEIVE macro. When this operand is omitted or coded NO the WKAREA operand is required.

WKAREA= Required entry when STATUS=NO. The WKAREA operand specifies a storage area for the received message and a six-word prefix field that is identical to the UTCB prefix field. This prefix field appears in the WKAREA, it always precedes the received message, and is updated with primary status information each time a transmission appears in the WKAREA. When defining the WKAREA storage field, the storage field length must be equal to or exceed the storage area defined by the TERMDEV macro TRANSIZ operand. To identify the terminal sending the message, the user must interrogate the work area's terminal name field after the request is completed. The data-name address must start with an alphabetic character and be no longer than eight alphanumeric characters. If a register number specifies the address, the symbol @ must precede the register number. When this operand is omitted the STATUS operand must be coded YES.

RETURN= An optional entry that allows the user to continue processing after issuing the RECEIVE macro. If this operand is omitted or coded NO, the user's program must wait until the Complete bit is set in the RECEIVE macro's parameter packet before processing continues.

PREFIX= An optional entry. If PREFIX=YES is specified, a single terminal operation is being requested and the terminal name must be in the UTCB prefix current terminal name field. PREFIX=NO is assumed if the operand is omitted. If PREFIX is specified, it must be used both with LIST=YES and LIST=NO.

LIST= An optional entry that specifies whether the macro expansion is to generate the service request and/or the parameter packet.

If LIST=YES, only the parameter packet is generated. Other RECEIVE operands allowable with LIST=YES are:

```
[TERMNAM=name of TERMINAL macro]
[,WKAREA={@Rn      }]
         {dataname }
[,STATUS={NO }]
         {YES}
[,PREFIX={NO }]
         {YES}
,LIST=YES
```

If LIST=NO, only the service request is generated. All of the RECEIVE operands except STATUS can be used with LIST=NO.

If LIST=null (no entry), both the service request and the parameter packet are generated. The default value for LIST is null.

LISTNAM= A required entry if LIST=NO is used. This operand specifies the address of the parameter packet associated with the service request.

TERMINATION CONDITIONS

Return Codes	Description
00	Normal completion.
D0	Terminal(s)/line not defined by Control Language.
D3	Terminal disabled.
D4	Macro usage error.
D5	Exception condition.
D6	BREAK or RVI received.
DC	Request not honored because job is in termination.

Primary Status Word Bit Number	Description
0-4	Reserved.
5	Transparent message received.
6	Reserved.
7	If 0 – EOU received If 1 – EOM received
8	Software time-out.
9	No termination character received.
10	BREAK or RVI received.
11	Transmission error. (The message will not be translated or edited.)
12	Modem error.
13	Line disconnected.
14	Hardware lost data.
15	Hardware time-out (synchronous only).

EXPANSION STORAGE REQUIREMENTS

To calculate the storage requirements for the RECEIVE macro, total the applicable storage quantities from the chart below.

Condition	Storage Requirements
The STATUS operand is selected and register notation omitted.	5 words
WKAREA operand is selected.	2 words
Data-name is addressed by register notation.	2 words
The terminal macro is addressed by register notation.	2 words

REMARKS

When a RECEIVE macro requesting status transfers status to the UTCB's Primary and Secondary Status fields, the status data refers to the hardware conditions of the terminal and the line at the time the receive status request is granted. The Primary Status word received at the completion of a receive data request, and residing in the work area prefix, refers to the hardware conditions of the terminal and line when the message residing in the work area was transferred from the terminal to the buffer.

The Secondary Status word, located in the UTCB, fills only after a STATUS=YES RECEIVE macro is issued. A brief description of the Secondary Status word bit designations is listed in following text. Appendix E contains an expanded description of the secondary status word.

Secondary Status Word Bit Number	Description
0	Terminal physically enabled.
1	Terminal logically enabled.
2	Terminal RECEIVE in progress.
3	Terminal SEND in progress.
4	Reserved.
5	Set when DISABLE is requested.
6	A RECEIVE just completed on the terminal specified in the UTCB prefix.
7	All output data queued for this terminal has been transmitted.
8-11	Reserved.
12-15	Terminal device type. B'0001' – Memorex terminal, not write only B'0010' – TTY 33, 35, 37, or 38 (ASR or KSR). B'0011' – Any synchronous processor that uses the line control specified in Appendix H. B'0101' – Memorex write only terminal.

EXAMPLE ONE – RECEIVE

Terminal 6 in the TRM678 TERMINAL macro is to be used. The user wants to check the status of terminal 6 before issuing a RECEIVE macro. The name of the terminal, TRM6, must be placed in the UTCB's prefix terminal name field before the macro is issued.

```
NAME      OPERATION   OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
          RECEIVE     TERMNAL=TRM678,STATUS=YES
```

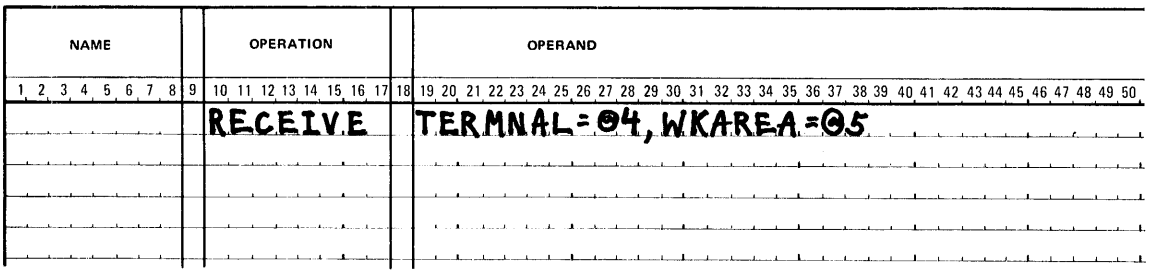
EXAMPLE TWO – RECEIVE

Assume that the status of terminal 6 was correct and the user wants to receive the first transmission from a buffer. Control is to be given to the user's program after the RECEIVE is initiated.

```
NAME      OPERATION   OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
          RECEIVE     TERMNAL=TRM678,RETURN=YES
```

EXAMPLE THREE – RECEIVE

Assume that terminal 8 in the TRM678 TERMINAL macro is used. The address of the UTCB generated by the TERMINAL macro is loaded into general register 4, and the work area address is loaded into general register 5. Both general registers were loaded before the RECEIVE macro was issued. Control returns to the applications program after a transmission appears in the work area.



SEND – WORK AREA TO BUFFER

The SEND macro requests the transfer of a message from the user's work area to the buffer and on to the destination terminal. The user has the option of using the SEND macro to space one line at the terminal, to allow TCOM to turn the line discipline to receive without an intervening send, or to not send previously queued data. Breaks sent by the terminal operator are honored and manipulated by the system as described under *Remarks.*

Name	Operation	Operands
[label]	SEND	TERMNAL={@R_n / name of TERMINAL macro} [,WKAREA=({data-name / @R_n} [{,integer / @R_n}])] [,MODE={UNIT / EOM / OK / STOP / NL / REWIND}] [,RETURN={NO / YES}] [,TRANSPT={NO / YES}] [,TAPE={NO / YES}] [,LIST={YES / NO / null}] [,LISTNAM={@R_n / data-name}]

NAME

A required entry if LIST=YES; otherwise, an optional entry. For label restrictions, refer to *Locating the Parameter Packet's Complete Bit and Return Code* in Chapter 5.

OPERANDS

TERMNAL= A required entry specifying the name of the TERMINAL macro defining the sending message's UTCB. A general register can be used to address the defining TERMINAL macro if the address is loaded into the general register before the RECEIVE macro is issued. The name of the TERMINAL macro must begin with an alphabetic character and be no longer than eight alphanumeric characters. If a register number specifies the TERMINAL macro's address, the symbol @ must precede the register number.

WKAREA= An optional entry. This operand may be omitted when the MODE operand's OK, STOP, or NL options are used. This operand must be used when the MODE=EOM or MODE=UNIT operand is used, or when the MODE operand is missing. The WKAREA operand specifies a storage area for the transmittable data, and a six-word prefix field that is identical to the UTCB's prefix field. This prefix field is always located in the work area field immediately before the transmittable data, and is updated with primary status information each time data is transmitted from the work area. The user identifies the receiving terminal by inserting the terminal name in the terminal name field of the work area prefix field before the SEND macro is issued. The user inserts the message length in the prefix field length field before issuing a SEND macro. The length must never exceed the longest buffer size integer defined by the //DEFINE statement.

The data-name operand must start with an alphabetic character and be no longer than eight alphanumeric characters. Any general registers may be used in place of the data-name providing the general registers are loaded with the storage address before the SEND is issued. When specifying register notation, the register number must be preceded by the @ symbol.

MODE= An optional entry. When omitted or if MODE=UNIT, the system assumes the user has divided his message into transmission units and wishes to notify the system that the last message sent was any unit other than the last unit of a segmented message.

The EOM entry specifies that the user has divided his transmissions into units and he wishes to notify the system that this transmission is the last or only unit of a segmented message. This operand is also used when messages are not segmented and the user notifies the system that the transmission is to be appended with an EOM control character.

The OK entry is only used when in the Conversational mode and the user has no message to send but wishes to receive from the terminal. Any transmission still in the buffers because the previous SEND to the terminal was broken will be transmitted to the terminal before the line is turned around. The user must identify the name of the terminal requiring the OK operation by placing the terminal name in the UTCB terminal name field before the SEND macro is issued.

The STOP entry terminates transmission to the terminal. Any buffer currently sending to a terminal will continue to completion. Subsequent buffers filled with data are ignored. After the stop is issued, the user may break the line discipline for one logical communication operation and issue a SEND or RECEIVE macro. The user must identify the terminal requiring the STOP operation by placing the terminal name in the UTCB terminal name field before the SEND macro is issued.

The NL entry is only used when communicating with asynchronous terminals; this entry resets the carriage at the beginning of the next line. The user must identify the terminal requiring the NL operation by placing the terminal name in the UTCB terminal name field before the SEND macro is issued. The NL operation is defined by the terminal's TRMDEV macro FFRAME, NEWLINE, and NULLS operands.

The REWIND entry is used to rewind the auxiliary device (for instance, MRX 1280 tape).

RETURN= An optional entry that allows the user to continue processing after issuing the SEND macro. If this operand is omitted or if RETURN=NO, the user's program must wait until the Complete bit is set in the SEND macro parameter packet before processing continues.

TRANSPT= An optional entry. YES specifies transparent mode transmission. The default is NO.

TAPE= An optional entry. YES specifies output on the auxiliary tape device. YES is required when operating a MRX 1280 at 120 cps; all output is directed to tape and the first 14 lines will be printed. Logical TCOM does not null-fill these first 14 lines to guarantee correct minimum line lengths. The default is NO.

LIST= An optional entry that specifies whether the macro expansion is to generate the service request and/or the parameter packet.

If LIST=YES, only the parameter packet is generated. Other SEND operands allowable with LIST=YES are:

```
[TERMNAL=name of TERMINAL macro]
[,WKAREA={@Rn      }]
         {dataname }
[       (UNIT  )]
[       (EOM   )]
[,MODE= (OK    )]
[       (STOP  )]
[       (NL    )]
[       (REWIND)]
[,TAPE={NO }]
       {YES}
[,TRANSPT={NO }]
          {YES}
```
(UNIT and NO are underlined as defaults.)

If LIST=NO, only the service request is generated. Any of the other SEND operands, with the exception of TRANSPT and TAPE can be used with LIST=NO.

If LIST=null (no entry), both the service request and the parameter packet are generated. The default value for LIST is null.

LISTNAM= A required entry if LIST=NO is specified. This operand specifies the address of the parameter packet associated with the service request.

TERMINATION CONDITIONS

Return Codes	Description
00	Normal completion.
D0	Terminal(s)/line not defined by Control Language.
D1	No buffer available for output because there are none large enough for this message.
D3	Terminal disabled.
D4	Macro usage error.
D5	Exception condition.
D6	BREAK or RVI received.
D8	Not enough space at enable time.
D9	The terminal name asked for with //DEF ID=xxxxxx, was not named in a TRMDEV macro which built the TDT descriptions on disc.
DA	Incorrect //DEF card in CLS.
DB	A terminal with an invalid ID attempted to connect.
DC	Request not honored because job is in termination.
DD	Irrecoverable Block I/O disc read error when reading SYSIN, //DEF tables, or the descriptions from NUCLIB.

Primary Status Word Bit Number	Description
8	Software time-out.
10	BREAK or RVI received.
12	Modem error.
13	Line disconnect.
15	Hardware time-out (synchronous only).

At the completion of the logical SEND communication operation, the Primary Status word will be in the Primary Status field of the work area prefix if the WKAREA operand is used, or in the Primary Status field of the UTCB prefix if the WKAREA operand is omitted.

EXPANSION STORAGE REQUIREMENTS

The storage requirements for the SEND macro are calculated by totaling the applicable storage values in the chart below.

Condition	Storage Required
Register notation and the WKAREA operand is omitted.	5 words
WKAREA operand is used.	2 words
Data-name is addressed by register notation.	2 words
Integer is addressed by register notation.	2 words
Integer keyword is used.	4 words
TERMINAL macro is addressed by register notation.	2 words

REMARKS

When the TERMINAL macro defines the transmission as Send-only mode and the terminal user transmits a break during the transmission the current data being transmitted is immediately halted.

The broken data can be retransmitted to the terminal by the terminal operator (pressing the BREAK key on the terminal), or by the applications programmer issuing another SEND macro. If a SEND macro is issued, the broken transmission will be transmitted first unless the SEND macro's MODE operand equals STOP.

When the TERMINAL macro defines the transmission modes as send and receive, or conversational, and the terminal user transmits a break when the processor is sending data, the segment being transmitted is immediately halted and considered by Logical TCOM as being the last segment of a segmented message. The user then has the option of issuing a RECEIVE for data, a RECEIVE for status, or another SEND logical communication operation. If a RECEIVE operation is issued, control will not return to the user until the RECEIVE operation completes. If a SEND operation is issued, the broken transmission precedes the message to be transmitted by the operation, unless the SEND macro's MODE operand indicates otherwise.

When status and return code data are reported to the user, the data recovered refers to the current or preceding communication operations. Delays in reporting status are caused by Logical TCOM service request and buffer transfer techniques. Since the user depends on the return code and status data to detect a break, the error might not be able to detect a break condition until several later logical communications operations have taken place. Messages sent are never queued more than one deep.

EXAMPLE ONE – SEND

Terminal 3 in the TRM345 TERMINAL macro is in the Conversational mode. The user wishes to receive two successive transmissions from terminal 3.

```
NAME      OPERATION   OPERAND
          SEND        TERMNAL=TRM345,MODE=OK
```

EXAMPLE TWO – SEND

Terminal 3 in the TRM345 TERMINAL macro is in the Send mode and is ready to send the fourth and final message segment. The message is to be stored in the alternate work area defined by registers 3 and 6. Prior to issuing the macro, the user loaded register 6 with the storage address and register 3 with the length of the field. Control returns to the user's program after issuing the SEND macro.

```
NAME      OPERATION   OPERAND
          SEND        TERMNAL=TRM345,WKAREA=(@6,@3),;
                      MODE=EOM,RETURN=YES
```

7. PHYSICAL COMMUNICATIONS PROGRAMMING

INTRODUCTION

Physical TCOM provides the user with direct controls over the basic procedures required to transfer messages over the communication lines. Control is given to the user through macros that assign lines, reserve control block storage, request I/O service, and define the executable I/O command program. Additional facilities allow the user to place an executable time limit on any command program and halt any command program.

The OPEN and CLOSE macros provide initialization at the physical level interface; the PCB (Physical Command Block) and ABORT macros generate control blocks; the EXCP macro (Execute Command Program) requests communications service; and the COMMAND macros generate the command words which comprise the command programs (such as ENABLE, READA, BREAK) executed by the communications driver.

ITB SUPPORT

ITB support allows the user to send and receive long data transmissions broken up into shorter blocks. Block Check Character generation increases the probability of detecting errors. BCCs are transmitted immediately following the ITB character. An Error Status Byte occupies one byte in the input buffer. The ESB has two indicators – the BCC Parity Error (bit 0) and the Hardware Lost Data (bit 1).

USER RESPONSIBILITY AT THE PHYSICAL LEVEL

User responsibility at the physical level is divided into five areas. These areas deal with controlling the line, transferring the message, translating and editing the message, testing for a completed command program, and writing an error recovery procedure.

LINE CONTROL

Line control is performed by the OPEN, CLOSE and EXCP macros. The OPEN macro initializes the lines so that program can request line service, and the EXCP macro requests the execution of a command program on the opened lines. The request terminates when the command program reaches a normal or abnormal termination, or when the time limit placed on the command program is exhausted. The CLOSE macro releases the specified lines.

MESSAGE TRANSFER

Before data can be sent, the physical level user must assemble a properly framed message in his work area. This message must contain the necessary pad and formatting characters for line and remote terminal control. He must also provide message translation facilities if the receiving terminal does not respond to the EBCDIC code. Before data can be received, the user must allocate storage and determine if he wants to terminate the message when his buffer is full, when any termination control character is sensed, or when a select set of termination control characters is sensed.

A list of termination control characters may be found listed with the READA command macro description in *Chapter 8. Physical Communications Macros.* Message transfer occurs when the operating system acknowledges the request for service and executes the command program.

MESSAGE EDITING

Once the user receives a message it is his responsibility to examine the message field and edit the message.

TESTING FOR A COMPLETED COMMAND PROGRAM

The user may test for a completed command program by identifying the PCB controlling the command program and testing the PCB's Complete bit for a set condition. The PCB expansion is located in Appendix C.

ERROR RECOVERY PROCEDURES

At the physical level all error recovery routines must be written by the user. The user tests for transmission errors after the communications command program is completed. The test is performed by testing the error conditions in the PCB (Physical Command Block) for any error conditions. If these bytes indicate an error, the user must provide any error recovery routine or terminate the program. For a complete description of the termination conditions, refer to the PCB Return Codes and Status charts in Appendix C.

MAIN FLOW OF A PHYSICAL TELECOMMUNICATIONS SESSION

All physical level programs must follow a certain basic flow. The following paragraphs outline a basic approach a user could adopt when writing a communications program. Figure 7-1 outlines the main flow of a telecommunication session. By no means is this example complex, and the telecommunications user will recognize its failings. But by following this example and using all the telecommunications macros, the user should be able to expand this flow to meet his applications and system line and timing requirements.

At the start of a session, the user must select the line he wants to service by using the OPEN macro. After opening the lines, he uses the EXCP macro to notify the operating system that his program requests the execution of a command program. The operating system then schedules and executes the command program. The applications programmer should establish the habit of disabling a line before enabling it for the first time to prevent any errors arising from enabling a previously enabled line.

A non-switched line is serviced by using various combinations of the COMMAND macros that set, enable, read, write, and disable the line. When establishing an asynchronous line connection, the SET COMMAND macro should appear before the COMMAND macros responsible for reading, writing, or disabling the line. When establishing synchronous communications, the SET COMMAND macro is omitted. The SET COMMAND macro conditions the ICA to the line speed, code, and parity encountered during asynchronous data transfer, and the ENABLE COMMAND macro establishes program connection. The COMMAND macros responsible for reading and writing can now transfer messages between the remote points. After all messages have been transferred, the connection is severed by the DISABLE COMMAND macro. Once a line is set, it need not be set again unless it is disabled.

Switched-line servicing is similar to non-switched servicing. The DISABLE and, if required, the SET macros still appear, but the ENABLE or ANSWER macro must be used to make the line connection. If the user is addressing a manually dialed terminal, he issues the ENABLE COMMAND macro and then issues dialing instructions to the computer operator. The applications programmer must remember that it is his responsibility to notify the computer operator of the remote terminal's phone number and provide an alternate path if the connection cannot be established. If he is waiting to be called, he uses the ANSWER macro. The connection is complete after the dialed call is answered. Message transfer and line disabling occur as they existed in non-switched line servicing.

When the user is finished with the teleprocessing lines, he releases it from his control by using the CLOSE macro.

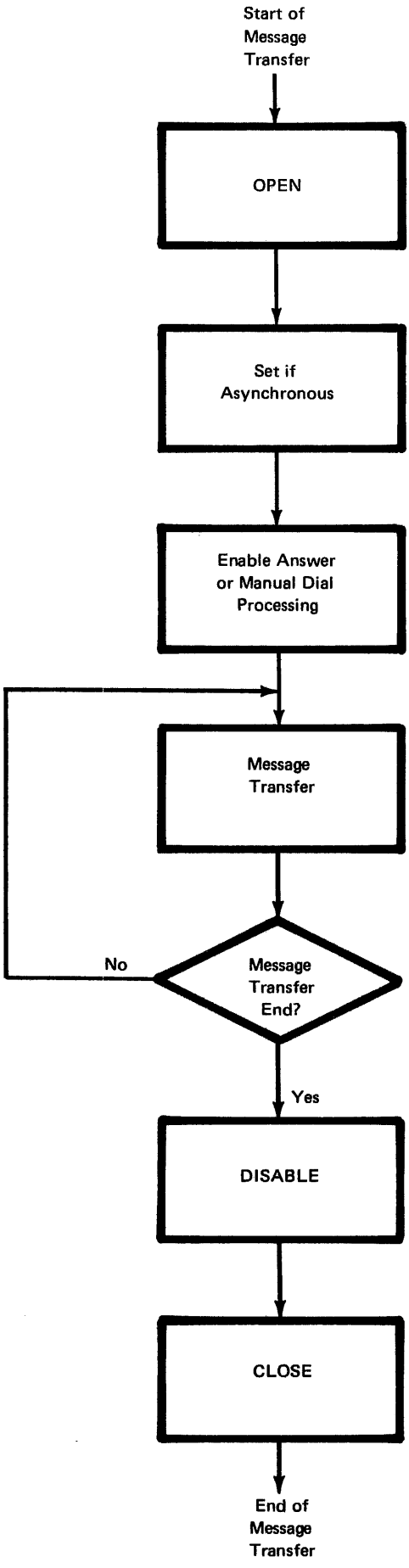


Figure 7-1. Main Flow of Physical Telecommunications Session

8. PHYSICAL COMMUNICATIONS MACROS

INTRODUCTION

The physical level macros are divided into four groups dealing with line initialization, service request, the physical command block, and teleprocessing commands. This section discusses these macros, shows their format, describes their termination conditions (when applicable), defines their storage requirements, and provides examples if necessary.

LINE INITIALIZATION MACROS

The line initialization macros assign lines to the user for use in his program (OPEN macro) and release lines from his program (CLOSE macro).

OPEN – OPEN LINE

The OPEN macro opens one line for processing; requests for line service cannot be honored until the line is opened.

Name	Operation	Operand
[label]	OPEN	IDENT=linename ,IOTYP=P ,BUFADR=symbolic address

NAME

An optional alphanumeric name. The first character of the name must be alphabetic, and the name cannot be longer than eight alphanumeric characters.

OPERANDS

IDENT= Specifies the line identifier as specified by the IDENT= operand of the MRX/OS Control Language //DEFINE statement.

IOTYP=P A required entry specifying physical-level line interface.

BUFADR= A required entry specifying an area in which information about the line (the unit ordinal) is to be returned.

TERMINATION CONDITIONS

Program termination may result from any one of the following conditions.

1. No //DEFINE statement in Control Language matching this IDENT.
2. No BUFADR specified.
3. Request completed normally.

EXPANSION STORAGE REQUIREMENTS

The OPEN macro requires four words of storage.

REMARKS

An expansion of this macro may be found in the **MRX/OS Control Program and Data Management Services, Basic Reference** manual.

EXAMPLE – OPEN

This example opens one line with the logical name L5 and inserts information about the line into BUFFER.

```
NAME                 OPERATION            OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
                     OPEN                 IDENT=L5,IOTYP=P,BUFADR=BUFFER
```

CLOSE – CLOSE LINE

The CLOSE macro releases the applications program from one line previously used for message transfer.

Name	Operation	Operand
[label]	CLOSE	IDENT=linename ,IOTYP=P

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight characters.

OPERANDS

IDENT= Specifies the logical name of the line to be closed. This name must not exceed eight characters and must be identical to the name used in the Control Language //DEFINE statement IDENT operand.

IOTYP=P A required entry specifying physical-level line interface.

EXPANSION STORAGE REQUIREMENTS

The CLOSE macro requires four words of storage.

REMARKS

The expansion for this macro may be found in the **MRX/OS Control Program and Data Management Services, Basic Reference** manual.

EXAMPLE – CLOSE

This example closes the line opened in the OPEN example (see immediately preceding text).

NAME	OPERATION	OPERAND
1 2 3 4 5 6 7 8	9 10 11 12 13 14 15 16 17	18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
	CLOSE	IDENT=L5,IOTYP=P

SERVICE REQUEST MACRO

EXCP – EXECUTE COMMAND PROGRAM

The EXCP macro requests the execution of a telecommunications command program on an opened line.

Name	Operation	Operand
[label]	EXCP	[PCB={@Rn / address}] [,CP={@Rn / address}] [,RETURN={NO / YES}] [,ERRCOMP={NO / YES}] [,UNORD={@Rn / address}]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

PCB= An optional entry specifying an address of a field whose contents identify the location of the PCB or ABORT macros. If omitted, the address must be in general register six before the request is made. Use of this operand causes the specified address to be loaded into general register six. The address option must start with an alphabetic character and be no longer than eight alphanumeric characters. Any general register containing the address may be used instead of the address option. The register number must be preceded by the symbol @.

CP= An optional entry specifying an address of a field whose contents identify the location of the command program. If this entry is used, the CPADR operand in the PCB macro is overridden. Omit this operand if the PCB operand addresses a field that holds the address of the command program to be executed. The address option must start with an alphabetic character and be no longer than eight alphanumeric characters. Any general register except register six may be used instead of the address option, provided that the address is loaded into the register before the EXCP macro is issued. The register number must be preceded by the symbol @.

RETURN= An optional entry. When coded RETURN=YES, control is returned to the program after I/O service is requested. When coded RETURN=NO, control is not returned to the program until the I/O operation is complete. If the operand is omitted, RETURN=NO is assumed.

ERRCOMP= If coded ERRCOMP=YES and the Abnormal bit in the PCB is set, it is the user's responsibility to branch to an exception processing error recovery routine. The PCB's Abnormal bit is set when the command program senses an invalid command or zero byte

count. If this option is coded ERRCOMP=NO, the applications program terminates when the command program encounters an invalid command or zero byte count. If the operand is omitted, ERRCOMP=NO is assumed.

UNORD= An optional entry. If coded, the unit ordinal will be properly supplied to the PCB before execution of the EXCP service request. The contents must have correspondence to the returned parameter from the OPEN macro.

TERMINATION CONDITIONS

If a user requests the execution of a command program or an Abort operation on a line that has not been opened, the Complete, Abnormal, and Return Code field addressed by the PCB will contain a hexadecimal C031. If a user requests the execution of an abort operation on an opened line, the Complete, Abnormal, and Return Code fields of the ABORT PCB will contain a hexadecimal 8000. Return codes for completed command programs are listed for each COMMAND macro.

EXPANSION STORAGE REQUIREMENTS

Keywords Selected	Storage Required
None	1 word
PCB=@R_n	2 words
PCB=address	3 words
CP=@R_n	3 words
CP=address	4 words
UNORD=@R_n	2 words
UNORD=address	3 words

REMARKS

The expansion for the EXCP macro can be found in the **MRX/OS Control Program and Data Management Services, Basic Reference** manual.

EXAMPLE ONE – EXCP

The user program requires telecommunications I/O service on an opened line. The address of the PCB is EX1 and has been previously loaded into INPCB. The command program address is found in the PCB macro and the user elects to continue processing after the operation completes. The command program terminates when an invalid command or zero byte count is sensed.

```
NAME                 OPERATION            OPERAND
1 2 3 4 5 6 7 8 | 9 | 10 11 12 13 14 15 16 17 | 18 | 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
                |   | EXCP                     |    | PCB=INPCB
```

EXAMPLE TWO – EXCP

The user program requires telecommunications I/O service on an opened line. The address of the PCB is EX2 and has been previously loaded into INPCB. The command program OVRD will override the command program referenced in the PCB macro and the address of the OVRD program has been loaded into INOVRD. Control is returned to the program after I/O service is requested. The user will write an exception processing routine.

```
NAME                 OPERATION            OPERAND
1 2 3 4 5 6 7 8 | 9 | 10 11 12 13 14 15 16 17 | 18 | 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
                |   | EXCP                     |    | PCB=INPCB,CP=INOVRD,RETURN=YES,;
                |   |                          |    | ERRCOMP=YES
```

PCB GENERATING MACROS

The PCB and ABORT macros are used to generate a physical command block expansion. A physical command block is used for all COMMAND and ABORT operations. The PCB macro generates a block that is seven words in length, and the ABORT macro generates a block that is three words in length.

PCB – GENERATE PHYSICAL COMMAND BLOCK

The PCB macro generates a physical command block, the main purpose of which is to provide the address of the command program. This macro produces no executable code.

Name	Operation	Operand
[label]	PCB	[CPADR=cp-name] [,DEVTYP=COMM]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

CPADR= A data-name pointing to the first COMMAND word to be executed. If this entry is omitted, it must appear in the EXCP macro. The data-name field must start with an alphabetic character and be no longer than eight alphanumeric characters.

DEVTYP=COMM An optional entry distinguishing this PCB as a telecommunications PCB rather than a PCB for control of other I/O devices. Different I/O devices require varying sizes of status fields. If this keyword is omitted, the PCB generated will be appended with three unused words.

EXPANSION STORAGE REQUIREMENTS

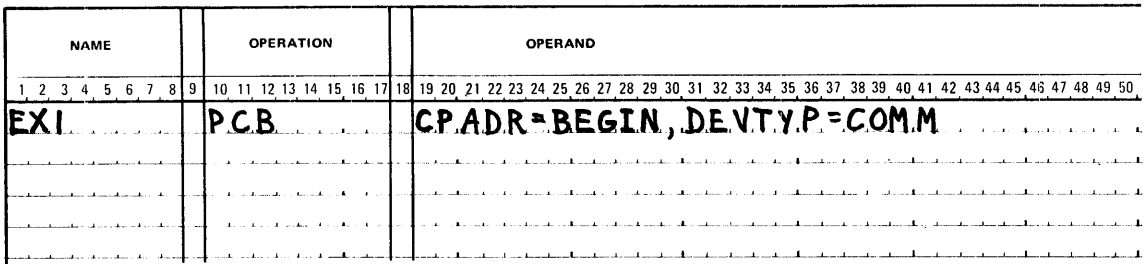
The PCB macro requires seven words of storage if the DEVTYP=COMM operand is used, or 10 words of storage if the DEVTYP operand is omitted.

REMARKS

The expansion for this macro can be found in Appendix C.

EXAMPLE – PCB

This example constructs a PCB block for the UPDT (update) line. The CP pointer in the EXCP macro is omitted. The command program starts in the Name Field labeled BEGIN. The PCB operand in the EXCP macro points to EX1.



ABORT – TERMINATE COMMAND PROGRAM

The ABORT macro generates a physical command block that requests the immediate line termination of the command program in process. After the system grants the request, the system flags the physical command block as complete and processing continues. This macro produces no executable code.

Name	Operation	Operand
[label]	ABORT	Blank

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

(none)

EXPANSION STORAGE REQUIREMENT

The ABORT macro requires three words of storage.

REMARKS

The expansion for this macro can be found in Appendix C. The CP operand in the EXCP macro must be omitted when requesting an ABORT operation.

COMMAND – TELEPROCESSING COMMAND MACROS

Each COMMAND macro constructs one command word. All COMMAND macros have a mandatory entry of COMMAND in the operation field; the particular type of COMMAND macro is determined by the OPCODE operand. Table 8-2 summarizes information about COMMAND macro operations. When more than one COMMAND macro is needed to complete message transfer, a command program is formed. The expansion for the command macros can be found in Appendix D.

USE OF THE TIMER=INTEGER OPERAND

The TIMER=integer operand is an optional entry that can be used to place a time limit on the execution of one or more COMMAND words. This operand may be used with one or more COMMAND macros in a command program. If a COMMAND macro omits this operand, TCOM does not place a time limit on the execution of the command. If the integer is zero, the amount of command execution time depends on the time remaining on the preceding command word's timer. If the integer is any value one through nine, TCOM places a time limit on the execution of the command. (Refer to Table 8-1.) When a command uses the timer option and time-out occurs, TCOM sets the Time-Out Status bit in the PCB and terminates the command program. When the SET, STATUS, CJE, CJNE, JSR, RTNJ and JUMP commands use the optional timer operand, timer expiration does not cause command program termination. However, if time-out occurs during the servicing of one of the seven mentioned commands, the command program terminates at the initiation of the next command in the command program. If the command program encounters a command using a reserved timer integer the command program will terminate and an invalid timer return code will be returned.

Table 8-1. Timer Integer and Time Limits

Timer Integer	Time Limit (in seconds)
0	Time remaining on preceding command word's timer.
1	1 second
2	3 seconds
3	15 seconds
4	30 seconds
5	60 seconds
6	180 seconds
7	300 seconds
8	600 seconds
9	1800 seconds

TERMINATION CONDITIONS

Termination conditions are posted in bytes 2, 3, 10, and 11 of the PCB addressing the command program being executed. The system posts these conditions when the command program terminates. A complete explanation of each possible command programming termination is provided in the PCB Return Codes and Status Codes charts in Appendix C. For the reader's convenience, a brief explanation of the possible termination conditions for a particular COMMAND macro is listed with each macro.

ENABLE – CONNECT LOCAL, NON-SWITCHED, OR SWITCHED MANUAL-DIAL LINE

This operation connects and validates the connection of a local, non-switched, or switched manual-dial line.*

Name	Operation	Operand
[label]	COMMAND	OPCODE=ENABLE [,CHAIN={NO / YES}] [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=ENABLE A required entry specifying the executable command.

CHAIN= An optional entry used to specify command chaining. The default is NO.

TIMER= An optional entry; refer to Table 8-1 for integer values.

*For the current TCOM release, ENABLE and ANSWER can be used interchangeably.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Connection established	80	00	00	00
Abort requested	80	00	80	00
Timer expired	80	00	40	00
Modem error	80	00	00	80
Invalid command	C0	C2	00	00
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

The ENABLE coded COMMAND macro requires one word of storage.

ANSWER – CONNECT SWITCHED ANSWER-ONLY LINE

This operation enables and connects switched answer-only lines.*

Name	Operation	Operand
[label]	COMMAND	OPCODE=ANSWER [,CHAIN={NO / YES}] [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=ANSWER A required entry specifying the executable command.

CHAIN= An optional entry used to specify command chaining. The default is NO.

TIMER= An optional entry; refer to Table 8-1 for integer values.

*For the current TCOM release, ANSWER and ENABLE can be used interchangeably.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Connection established	80	00	00	00
Abort requested	80	00	80	00
Timer expired	80	00	40	00
Modem error	80	00	00	80
Invalid command	C0	C2	00	00
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

The ANSWER coded COMMAND macro requires one word of storage.

REMARKS

The command may be issued in response to a RPTRING coded COMMAND macro or in anticipation of a ring indicator.

EXAMPLE – ANSWER

The receiving station will be expecting a call from a remote location in less than one minute. Another command is to be chained to the ANSWER coded COMMAND macro.

```
NAME      OPERATION      OPERAND
1 2 3 4 5 6 7 8 | 9 | 10 11 12 13 14 15 16 17 | 18 | 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
          COMMAND      OPCODE=ANSWER,CHAIN=YES,TIMER=5
```

RPTRING – REPORT RING INDICATOR

This operation reports the receipt of a ring indicator by terminating the command.

Name	Operation	Operand
[label]	COMMAND	OPCODE=RPTRING [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphanumeric character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=RPTRING A required entry specifying the executable command.

TIMER= An optional entry; refer to Table 8-1 for integer values.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Ring receive	80	00	00	00
Abort requested	80	00	80	00
Timer expired	80	00	40	00
Invalid command	C0	C2	00	00
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

The RPTRING coded COMMAND macro requires one word of storage.

EXAMPLE – RPTRING

NAME		OPERATION		OPERAND
1 2 3 4 5 6 7 8	9	10 11 12 13 14 15 16 17	18	19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
		COMMAND		OPCODE=RPTRING

DISABLE -- DISCONNECT ANY TYPE OF LINE

This operation disconnects and clears all types of lines.

Name	Operation	Operand
[label]	COMMAND	OPCODE=DISABLE [,CHAIN={NO / YES}] [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=DISABLE A required entry specifying the executable command.

CHAIN= An optional entry used to specify command chaining. The default is NO.

TIMER= An optional entry; refer to Table 8-1 for integer values.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Connection broken	80	00	00	00
Abort requested	80	00	80	00
Timer expired	80	00	40	00
Invalid timer	C0	C6	00	00

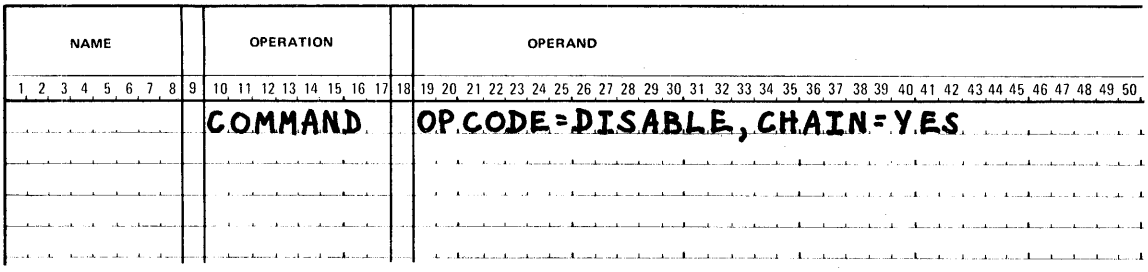
EXPANSION STORAGE REQUIREMENTS

The DISABLE coded COMMAND macro requires one word of storage.

REMARKS

For switched lines, commands chained to (or requested after) a DISABLE command will be held for three seconds prior to execution.

EXAMPLE – DISABLE



JUMP – COMMAND PROGRAM JUMP TO SPECIFIED ADDRESS

This operation causes command program execution to transfer and continue at the address specified.

Name	Operation	Operand
[label]	COMMAND	OPCODE=JUMP ,CWADR=data-name [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=JUMP A required entry specifying the executable command code.

CWADR= Specifies the symbolic address of the next executable command word in the command program.

TIMER= An optional entry; refer to Table 8-1 for integer values.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

The JUMP coded COMMAND macro requires three words of storage.

READA -- MOVE INCOMING DATA INTO STORAGE

This operation causes data arriving on the line to be put in storage beginning at the address specified by the BUFADR operand. Control characters received by the hardware are examined to see if they match the termination control characters appearing in Table 8-2. When a termination character is sensed, the command program terminates. If a termination control character is not recognized and the number of bytes received equals BUFSIZ, the READA command terminates and processing of the command program continues. When in the synchronous mode, all pad characters will be stripped from the received message before the message enters the BUFADR area if a hardware recognizable control character is received. Otherwise, pad characters will appear in the read message if the data read field is shorter than the buffer area (BUFSIZ).

Name	Operation	Operands
[label]	COMMAND	OPCODE=READA ,BUFADR=data-name ,BUFSIZ=integer [,CHAIN={NO / YES}] [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=READA A required entry specifying the executable command code.

BUFADR= Specifies the symbolic address of the storage locations containing the received field. The data-name field must start with an alphabetic character and be no longer than eight alphanumeric characters.

BUFSIZ= The number of storage locations reserved for the input message. The length of the input message cannot exceed the available storage. When using ITB, one byte must be allowed for each Error Status Byte* in the input buffer.

CHAIN= An optional entry used to specify command chaining. The default is NO.

TIMER= An optional entry; refer to Table 8-1 for integer values.

*The ESB contains a BCC parity error indicator (bit 1) and a Hardware lost data indicator (bit 0). If one SYN character does not follow the BCC after ITB, the ESB will be placed in the receiving buffer following the first character of the next block.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Buffer exhausted	80	00	00	00
Abort requested	80	00	80	00
Timer expired	80	00	40	00
Termination character received	80	00	20	00
Intermediate BCC detected	80	00	08	00
Software lost data	80	00	04	00
Transmission error	80	00	02	00
Modem error	80	00	00	80
Line disconnect detected	80	00	00	40
Hardware lost data	80	00	00	04
Hardware timer expired	80	00	00	02
Invalid command	C0	C2	00	00
Zero byte count	C0	C4	00	00
Invalid timer	C0	C6	00	00

If the command program terminates during READA because of an ABORT request or an expired timer, the line will be left in receive mode. If the READA terminates because of a full buffer (residual byte count = 0) containing no termination characters, and there is a command word chained to the command program, the line will be left in Receive mode; otherwise, the line will be put in Transmit mode.

Table 8-2. Termination Control Characters

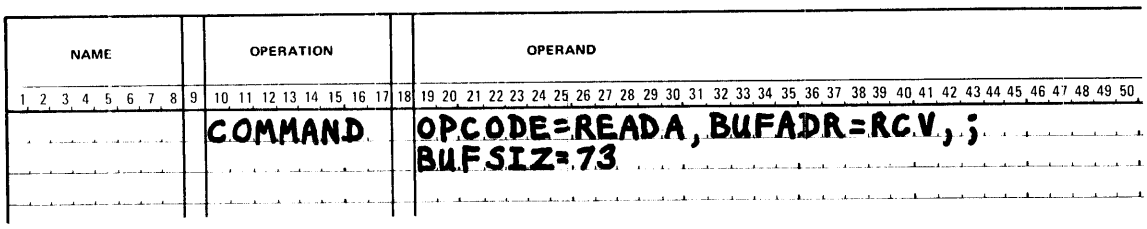
Termination Control Characters	EBCDIC	ASCII
EOT	37	04
ETX	03	03
ETB/EOB	26	17
CR	0D	0D
ENQ	2D	05
NAK	3D	15
CAN	18	18

EXPANSION STORAGE REQUIREMENTS

The READA coded command macro requires four words of storage.

EXAMPLE – READA

A message of less than 73 characters will be received and stored at address RCV. Another COMMAND macro appears in the command program being executed.



READA1* – MOVE INCOMING DATA INTO STORAGE

This operation causes data arriving on the line to be put in storage beginning at the address specified by the BUFADR operand. The READA1 command operates the same as the READA command with the exception that an additional character, the character following the termination control character, is brought into storage.

Name	Operation	Operand
[label]	COMMAND	OPCODE=READA1 ,BUFADR=data-name ,BUFSIZ=integer [,CHAIN={NO / YES}] [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=READA1 A required entry specifying the executable command code.

BUFADR= Specifies the symbolic address of the storage locations containing the received field. The data-name field must start with an alphabetic character and be no longer than eight alphanumeric characters.

*Recommended for asynchronous devices.

BUFSIZ= The number of storage locations reserved for the input message. The length of the input message cannot exceed the available storage. For ITB input, one byte must be allowed for each Error Status Byte in the input buffer. If one SYN character does not follow the BCC after ITB, the ESB will be placed in the receiving buffer following the first character of the next block.

CHAIN= An optional entry used to specify command chaining. The default is NO.

TIMER= An optional entry; refer to Table 8-1 for integer values.

EXPANSION STORAGE REQUIREMENTS

The READA1 coded COMMAND macro requires four words of storage.

EXAMPLE – READA1

```
NAME      OPERATION    OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
          COMMAND      OPCODE=READA1,BUFADR=RCV,;
                       BUFSIZ=73
```

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Buffer exhausted	80	00	00	00
Abort requested	80	00	80	00
Timer expired	80	00	40	00
Termination character received	80	00	20	00
Intermediate BCC detected	80	00	08	00
Software lost data	80	00	04	00
Transmission error	80	00	02	00
Modem error	80	00	00	80
Line disconnect detected	80	00	00	40
Hardware lost data	80	00	00	04
Hardware timer expired	80	00	00	02
Invalid command	C0	C2	00	00
Zero byte count	C0	C4	00	00
Invalid timer	C0	C6	00	00

If the command program terminates during READA1 because of an ABORT request or an expired timer, the line will be left in receive mode. If the READA1 terminates because of a full buffer (residual byte count = 0) containing no termination characters, and there is a command word chained to the command program, the line will be left in Receive mode; otherwise, the line will be put in Transmit mode.

READS – MOVE INCOMING DATA INTO STORAGE

This operation causes data arriving on the line to be put in storage beginning at the address specified by the BUFADR operand. Control characters received by the hardware are compared against the termination control characters appearing in the ENDLIST. When a match occurs the command program terminates. If no match occurs and the number of bytes received equals BUFSIZ, the READS command terminates and processing of the command program continues. When in the synchronous mode all pad characters will be stripped from the received message before the message enters the BUFADR area if a hardware recognizable control character defined by the ENDLIST operand is received. Otherwise, pad characters will appear in the read message if the read field is shorter than the buffer area (BUFSIZ).

Name	Operation	Operand
[label]	COMMAND	OPCODE=READS ,BUFADR=data-name ,BUFSIZ=integer ,ENDLIST=(xx,xx,xx,xx) [,CHAIN={NO / YES}] [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=READS A required entry specifying the executable command.

BUFADR= Specifies the symbolic storage address of the receiving field. The data-name must start with an alphabetic character and be no longer than eight alphanumeric characters.

BUFSIZ= The number of storage locations reserved for the input message. The length of the input message cannot exceed the available storage. When using ITB, one byte must be allowed for each Error Status Byte in the input buffer.

ENDLIST= One to four termination control characters to be used as message termination control characters (refer to Table 8-2 in immediately preceding text).

CHAIN= An optional entry used to specify command chaining. The default is NO.

TIMER= An optional entry; refer to Table 8-1 for integer values.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Buffer exhausted	80	00	00	00
Abort requested	80	00	80	00
Timer expired	80	00	40	00
Termination character received	80	00	20	00
Block check character detected	80	00	08	00
Software lost data	80	00	04	00
Transmission error	80	00	02	00
Modem error	80	00	00	80
Line disconnect detected	80	00	00	40
Hardware lost data	80	00	00	04
Hardware timer expired	80	00	00	02
Invalid command	C0	C2	00	00
Invalid timer	C0	C6	00	00
Zero byte count	C0	C4	00	00

EXPANSION STORAGE REQUIREMENTS

The READS coded COMMAND macro requires six words of storage.

EXAMPLE – READS

An EBCDIC message of less than 161 characters will be received and stored at address RCV1 in less than three minutes. The message will terminate on any one of three characters appearing in the command terminator list. Another command appears in the command program being executed.

```
NAME      OPERATION   OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
          COMMAND     OPCODE=READS,BUFADR=RCV1,BUFSIZ;
                      =160,ENDLIST=(26,2D,37),CHAIN=;
                      YES,TIMER=6
```

OPERANDS

OPCODE=READS1 A required entry specifying the executable command.

BUFADR= Specifies the symbolic storage address of the receiving field. The data-name must start with an alphabetic character and be no longer than eight alphanumeric characters.

BUFSIZ= The number of storage locations reserved for the input message. The length of the input message cannot exceed the available storage. When using ITB, one byte must be allowed for each Error Status Byte in the input buffer.

ENDLIST= One to four termination control characters to be used as message termination control characters (refer to Table 8-2 in immediately preceding text).

CHAIN= An optional entry used to specify command chaining. The default is NO.

TIMER= An optional entry; refer to Table 8-1 for integer values.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Buffer exhausted	80	00	00	00
Abort requested	80	00	80	00
Timer expired	80	00	40	00
Termination character received	80	00	20	00
Block check character detected	80	00	08	00
Software lost data	80	00	04	00
Transmission error	80	00	02	00
Modem error	80	00	00	80
Line disconnect detected	80	00	00	40
Hardware lost data	80	00	00	04
Hardware timer expired	80	00	00	02
Invalid command	C0	C2	00	00
Invalid timer	C0	C6	00	00
Zero byte count	C0	C4	00	00

READS1 – MOVE INCOMING DATA INTO STORAGE

This operation causes data arriving on the line to be put in storage beginning at the address specified by the BUFADR operand. Control characters received by the hardware are compared against the termination control characters appearing in the ENDLIST. When a match occurs the command program terminates. One more character, in addition to the termination character, is brought in after the termination character. If no match occurs and the number of bytes received equals BUFSIZ, the READS1 command terminates and processing of the command program continues. When in the synchronous mode all pad characters will be stripped from the received message before the message enters the BUFADR area if a hardware recognizable control character defined by the ENDLIST operand is received. Otherwise, pad characters will appear in the read message if the read field is shorter than the buffer area (BUFSIZ).

Name	Operation	Operand
[label]	COMMAND	OPCODE=READS1 ,BUFADR=data-name ,BUFSIZ=integer ,ENDLIST=(xx,xx,xx,xx) [,CHAIN={NO / YES}] [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

EXPANSION STORAGE REQUIREMENTS

The READS1 coded COMMAND macro requires six words of storage.

EXAMPLE – READS1

NAME	OPERATION	OPERAND
1 2 3 4 5 6 7 8 9	10 11 12 13 14 15 16 17 18	19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
	COMMAND	OPCODE=READS1,BUFADR=RCV1,;
		BUFSIZ=160,ENDLIST=(26,2D,37),;
		CHAIN=YES,TIMER=6

READN – MOVE INCOMING DATA INTO STORAGE

This operation causes data arriving on the line to be put in storage beginning at the address specified by the BUFADR operand. The command terminates when the number of bytes received equals the BUFSIZ operand. At termination, control reverts to the next command in the command program if one exists.

Name	Operation	Operand
label	COMMAND	OPCODE=READN ,BUFADR=data-name ,BUFSIZ=integer [,CHAIN={NO / YES}] [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphanumeric character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=READN A required entry specifying the executable COMMAND macro.

BUFADR= A required entry specifying the storage address of the received message. The data-name must start with an alphabetic character and be no longer than eight alphanumeric characters.

BUFSIZ= Number of storage locations reserved for the input message. The length of the input message cannot exceed the available storage. For ITB input, one byte must be allowed for each Error Status Byte in the input buffer.

CHAIN= An optional entry used to specify command chaining. The default is NO.

TIMER= An optional entry; refer to Table 8-1 for integer values.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Buffer exhausted	80	00	00	00
Abort requested	80	00	80	00
Timer expired	80	00	40	00
Intermediate BCC detected	80	00	08	00
Software lost data	80	00	04	00
Transmission error	80	00	02	00
Modem error	80	00	00	80
Line disconnect detected	80	00	00	40
Hardware lost data	80	00	00	04
Hardware timer expired	80	00	00	02
Invalid command	C0	C2	00	00
Zero byte count	C0	C4	00	00
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

The READN coded COMMAND macro requires four words of storage.

EXAMPLE – READN

A 120-character field of data will be received and stored in address RCV2. The message must arrive in less than one minute from the start of the command. Commands are chained.

NAME	OPERATION	OPERAND
	COMMAND	OPCODE=READN,BUFADR=RCV2,BUFSIZ; =120,CHAIN=YES,TIMER=5

WRITE – MOVE DATA FROM STORAGE TO THE LINE

This operation transmits the specified number of characters from the data buffer to the addressed communications line. All characters are treated as data characters. TCOM reports any breaks received during asynchronous transmission in the PCB's status field and transmission continues.

Name	Operation	Operands
[label]	COMMAND	OPCODE=WRITE ,BUFADR=data-name ,BUFSIZ=integer [,CHAIN={NO / YES}] [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=WRITE A required entry specifying the executable command.

BUFADR= A required entry specifying the storage address of the data. The data-name must start with an alphabetic character and be no longer than eight alphanumeric characters.

BUFSIZ= The number of bytes to be transmitted. The length of the transmitted message cannot exceed available storage.

CHAIN= An optional entry used to specify command chaining. The default is NO.

TIMER= An optional entry; refer to Table 8-1 for integer values.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Buffer exhausted	80	00	00	00
Abort requested	80	00	80	00
Timer expired	80	00	40	00
Break received	80	00	10	00
Software lost data	80	00	04	00
Modem error	80	00	00	80
Line disconnect detected	80	00	00	40
Hardware lost data	80	00	00	04
Hardware timer expired	80	00	00	02
Invalid command	C0	C2	00	00
Byte count zero	C0	C4	00	00
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

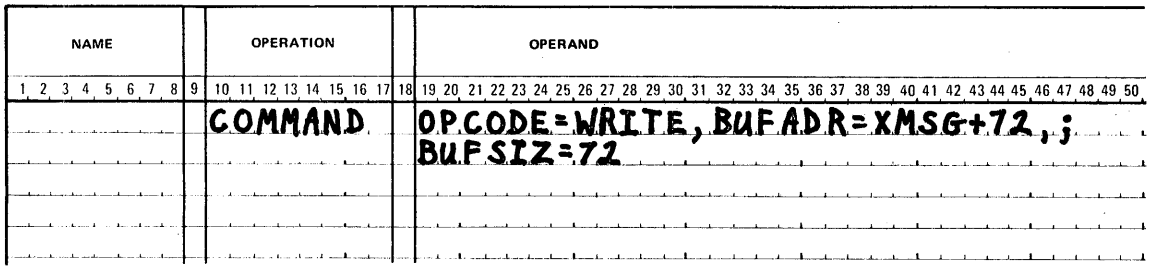
The WRITE coded COMMAND macro requires four words of storage.

REMARKS

Trailing pad characters for binary synchronous equipment must be supplied by the user immediately following the ETB or ETX character.

EXAMPLE – WRITE

Two 72-character messages appear in storage at address location XMSG. The second message in storage is to be transmitted. Commands in the command program are not to be chained.



REMARKS

Each message must be properly framed and coded in the BUFADR message area. Trailing pad characters for synchronous equipment must be supplied by the user immediately following the end frame (ETB or ETX) character.

WRITET – MOVE DATA FROM STORAGE TO THE LINE

This operation transmits the specified number of characters from the data buffer to the addressed communications line. All control characters transferred are treated as data characters. A break received during transmission terminates the command program before buffer exhaustion.

Name	Operation	Operands
[label]	COMMAND	OPCODE=WRITET ,BUFADR=data-name ,BUFSIZ=integer [,CHAIN={NO / YES}] [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=WRITET A required entry specifying the executable command.

BUFADR= A required entry specifying the storage address of the data to be transmitted. The data-name must start with an alphabetic character and be no longer than eight alphanumeric characters.

BUFSIZ= The number of bytes to be transmitted.

CHAIN= An optional entry used to specify command chaining. The default is NO.

TIMER= An optional entry; refer to Table 8-1 for integer values.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Buffer exhausted	80	00	00	00
Abort requested	80	00	80	00
Timer expired	80	00	40	00
Break received	80	00	10	00
Software lost data	80	00	04	00
Modem error	80	00	00	80
Line disconnect detected	80	00	00	40
Hardware lost data	80	00	00	04
Invalid command	C0	C2	00	00
Byte count zero	C0	C4	00	00
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

The WRITET coded COMMAND macro requires four words of storage.

REMARKS

TCOM prohibits the use of this command when transmitting synchronous data.

WRITEC – MOVE DATA FROM STORAGE TO THE LINE

This operation transmits data in the buffer to a synchronous communications line. Output characters are treated as transparent control characters; a DLE character is sent preceding the transmission of each character in the output buffer.

Name	Operation	Operand
[label]	COMMAND	OPCODE=WRITEC ,BUFADR=data-name ,BUFSIZ=integer [,TIMER=integer] [,CHAIN={NO / YES}]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=WRITEC A required entry specifying the executable command.

BUFADR= A required entry specifying the storage address of the data to be transmitted. The data-name must start with an alphabetic character and be no longer than eight alphanumeric characters.

BUFSIZ= The number of bytes in the message.

CHAIN= An optional entry used to specify command chaining. The default is NO.

TIMER= An optional entry; refer to Table 8-1 for integer values.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Buffer exhausted	80	00	00	00
Abort requested	80	00	80	00
Timer expired	80	00	40	00
Software lost data	80	00	04	00
Modem error	80	00	00	80
Line disconnect detected	80	00	00	40
Hardware lost data	80	00	00	04
Hardware timer expired	80	00	00	02
Invalid command	C0	C2	00	00
Byte count zero	C0	C4	10	00
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

The WRITEC coded COMMAND macro requires four words of storage.

RESYN – RESYNCHRONIZATION

This operation causes the re-establishment of character framing when the next synchronizing sequence is received.

Name	Operation	Operand
[label]	COMMAND	OPCODE=RESYN [,CHAIN={NO / YES}] [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=RESYN A required entry specifying the executable command.

CHAIN= An optional entry used to specify command chaining. The default is NO.

TIMER= An optional entry; refer to Table 8-1 for integer values.

TERMINATION CONDITIONS

Description	PCB Byte Indicators			
	2	3	10	11
Abort requested	80	00	80	00
Timer expired	80	00	40	00
Line disconnect detected	80	00	00	40
Invalid unit ordinal	C0	30	00	00
Invalid command	C0	C2	00	00
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

The RESYN coded COMMAND macro requires one word of storage.

EOTSCH – EOT SEARCH*

This operation causes a search for an EOT character. The command completes when EOT is found, abort is requested, disconnect is detected, or the command times out.

Name	Operation	Operand
[label]	COMMAND	OPCODE=EOTSCH [,CHAIN={NO / YES}] [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=EOTSCH A required entry specifying the executable command.

CHAIN= An optional entry used to specify command chaining. The default is NO.

TIMER= An optional entry; refer to Table 8-1 for integer values.

*This command is intended for use only when the Memorex central processing unit is a tributary station on a synchronous multi-point line that is half-duplex or full-duplex with SYN fill between transmissions. EOTSCH does not function properly on a synchronous, full-duplex, multi-point line with mark-hold between transmissions.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Abort requested	80	00	80	00
Timer expired	80	00	40	00
Line disconnect detected	80	00	00	40
Invalid unit ordinal	C0	30	00	00
Invalid command	C0	C2	00	00
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

The EOTSCH coded COMMAND macro requires one word of storage.

EXAMPLE – EOTSCH

The following example illustrates a command program that makes use of the EOTSCH command. EOTSCH does not work properly on a synchronous, full duplex, multi-point line with mark-hold between transmissions. In that case, the command sequence in brackets must be substituted for the first four command words.

```
NAME      OPERATION   OPERAND
EOTBUF    BRS         2
EOT       BDD         X'37FF'
ADRBUF    BRS         3
ENQ       BDD         X'2D'
POLL      BDD         X'C2F6'
SELECT    BDD         X'82F1'
SCHEOT    COMMAND     OPCODE=EOTSCH,CHAIN=YES
          COMMAND     OPCODE=READN,BUFSIZ=1,BUFADR=;
                      EOTBUF+1,CHAIN=YES
          COMMAND     OPCODE=CJE,BUFSIZ=1,BUFADR1=;
                      EOT+1,BUFADR2=EOTBUF+1,CWADR=;
                      RDADR,CHAIN=YES
          COMMAND     OPCODE=JUMP,CWADR=SCHEOT
RDEOT     COMMAND     OPCODE=READN,BUFSIZ=2,BUFADR1=;
                      EOT,BUFADR2=EOTBUF,CWADR=RDADR,;
                      CHAIN=YES
SCHEOT    COMMAND     OPCODE=RESYN,CHAIN=YES
          COMMAND     OPCODE=JUMP,CWADR=RDEOT
SCHADR    COMMAND     OPCODE=RESYN,CHAIN=YES
RDADR     COMMAND     OPCODE=READN,BUFSIZ=2,BUFADR=;
                      ADRBUF,CHAIN=YES
          COMMAND     OPCODE=CJE,BUFSIZ=2,BUFADR1=;
                      EOT,BUFADR=ADRBUF,CWADR=RDADR,;
                      CHAIN=YES
          COMMAND     OPCODE=READN,BUFSIZ=1,BUFADR=;
                      ADRBUF+2,CHAIN=YES
          COMMAND     OPCODE=CJNE,BUFSIZ=1,BUFADR=ENQ,;
                      BUFADR2=ADRBUF+2,CWADR=SCHEOT,;
                      CHAIN=YES
          COMMAND     OPCODE=CJE,BUFSIZ=2,BUFADR1=POLL;
                      ,BUFADR2=ADRBUF,CWADR=POLLED,;
                      CHAIN=YES
          COMMAND     OPCODE=CJNE,BUFSIZ=2,BUFADR1=;
                      SELECT,BUFADR2=ADRBUF,CWADR=;
                      SCHADR,CHAIN
SELECTED  COMMAND     OPCODE=WRITE
                      .
                      .
                      .
POLLED    COMMAND     OPCODE=WRITE
                      .
                      .
                      .
```

CJE – COMPARE AND JUMP IF EQUAL

The CJE operation compares up to 31 bytes of data in two fields. If the fields match, command execution continues at the jump address specified. If the fields do not match, command execution continues at the next command if command chaining has been specified. Execution of more than four consecutive CJE commands may result in hardware lost data if all the attached communications lines are transmitting simultaneously.

Name	Operation	Operand
[label]	COMMAND	OPCODE=CJE ,BUFSIZ=integer ,BUFADR1=data-name ,BUFADR2=data-name ,CWADR=data-name [,CHAIN={NO / YES}] [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=CJE A required entry specifying the executable command.

BUFSIZ= A required entry specifying the number of bytes in each buffer to be compared. This number must be ≤ 31.

BUFADR1= A required entry specifying the symbolic address of the beginning of the first field to be compared.

BUFADR2= A required entry specifying the symbolic address of the beginning of the second field to be compared.

CWADR= A required entry specifying the symbolic address of the next command to be executed if the two fields match.

CHAIN= An optional entry used to specify command chaining. The default is NO.

TIMER= An optional entry; refer to Table 8-1 for integer values.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

The CJE coded COMMAND macro requires six words of storage.

CJNE – COMPARE AND JUMP IF NOT EQUAL

The CJNE operation compares up to 31 bytes of data in two fields. If the fields do not match, command execution continues at the jump address specified. If the fields do match, command execution continues at the next command if the chain bit is on. Execution of more than four consecutive CJNE commands may result in hardware lost data if all the attached communications lines are transmitting simultaneously.

Name	Operation	Operand
[label]	COMMAND	OPCODE=CJNE ,BUFSIZ=integer ,BUFADR1=data-name ,BUFADR2=data-name ,CWADR=data-name [,CHAIN={NO / YES}] [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=CJNE A required entry specifying the executable command.

BUFSIZ= A required entry specifying the number of bytes in each buffer to be compared. This number must be ≤ 31.

BUFADR1= A required entry specifying the symbolic address of the beginning of the first field to be compared.

BUFADR2= A required entry specifying the symbolic address of the beginning of the second field to be compared.

CWADR= A required entry specifying the symbolic address of the next command to be executed if the two fields do not match.

CHAIN= An optional entry used to specify command chaining. The default is NO.

TIMER= An optional entry; refer to Table 8-1 for integer values.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

The CJNE coded COMMAND macro requires six words of storage.

JSR – JUMP TO SUBROUTINE

This operation provides the linkage for jumping to and executing a subroutine command program. After executing the subroutine, control can be returned to the command following the JSR by using the RTNJ command. JSR commands cannot be nested.

Name	Operation	Operand
[label]	COMMAND	OPCODE=JSR ,CWADR=data-name [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=JSR A required entry specifying the executable command.

CWADR= A required entry specifying the symbolic address of the next command word to be executed (the subroutine).

TIMER= An optional entry; refer to Table 8-1 for integer values.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

The JSR coded COMMAND macro requires three words of storage.

RTNJ – RETURN JUMP

This operation affects processing of the command word immediately following the JSR, causing the execution of the subroutine terminated by RTNJ.

Name	Operation	Operand
[label]	COMMAND	OPCODE=RTNJ [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=RTNJ A required entry specifying the executable command.

TIMER= An optional entry; refer to Table 8-1 for integer values.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

The RTNJ coded COMMAND macro requires one word of storage.

BREAK – TRANSMIT SPACE SIGNAL TO THE LINE

This operation causes one continuous space of at least 200 ms to be transmitted to the

Name	Operation	Operand
[label]	COMMAND	OPCODE=BREAK [,CHAIN={NO / YES}] [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=BREAK A required entry specifying the executable command.

CHAIN= An optional entry used to specify command chaining. The default is NO.

TIMER= An optional entry; refer to Table 8-1 for integer values.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Break transmitted	80	00	00	00
Timer expired	80	00	40	00
Abort requested	80	00	80	00
Modem error	80	00	00	80
Line Disconnect	80	00	00	40
Invalid command	C0	C2	00	00
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

The BREAK coded COMMAND macro requires one word of storage.

RCVBRK – RECOGNIZE RECEIPT OF SPACE SIGNAL

The RCVBRK operation recognizes the arrival of a 150 ms continuous space signal.

Name	Operation	Operand
[label]	COMMAND	CODE=RCVBRK [,TIMER=integer]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERAND

OPCODE=RCVBRK A required entry specifying the executable command.

TIMER= An optional entry; refer to Table 8-1 for integer values.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Abort requested	80	00	80	00
Timer expired	80	00	40	00
Break received	80	00	10	00
Software lost data	80	00	04	00
Modem error	80	00	00	80
Line disconnect	80	00	00	40
Hardware lost data	80	00	00	04
Invalid command	C0	C2	00	00
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

The RCVBRK coded COMMAND macro requires one word of storage.

SET – CONDITION ICA

This operation conditions the ICA to the line speed, code and parity used on the line during message transfer and can be issued at any time.

Name	Operation	Operand
[label]	COMMAND	OPCODE=SET [,SPEED={110 / 150 / 300 / 600}] [,LINECOD={ASCNO / ASCEVEN / ASCODD / EBCDIC}] [,CHAIN={NO / YES}] [,TIMER=integer] [,CPUXMIT={PRI / SEC}]

NAME

An optional alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=SET A required entry specifying the executable command.

SPEED= An optional entry specifying the BPS transfer rate. When omitted, the line defaults to 110 if the line can operate at 110, or 600 if it cannot.

LINECOD= An optional entry specifying the message line code and parity. ASCNO symbolizes the USASCII code without parity. ASCEVEN symbolizes the USASCII code with even parity. ASCODD symbolizes the USASCII code with odd parity. EBCDIC symbolizes the EBCDIC code without parity. If no entry is made, system defaults to ASCEVEN.

CHAIN= An optional entry used to specify command chaining. The default is NO.

TIMER= An optional entry; refer to Table 8-1 for integer values.

CPUXMIT= An optional entry specifying the channel to be used by the central processing unit for transmitting when operating with split speed hardware. PRI symbolizes the primary channel and SEC indicates the secondary channel. The default value is PRI.

TERMINATION CONDITIONS

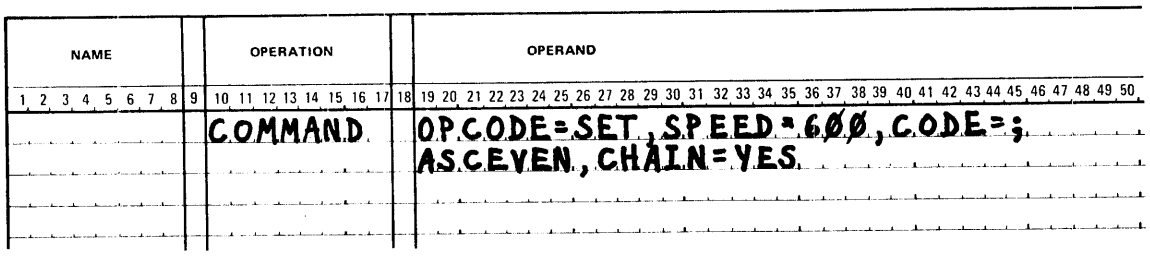
Description	PCB Byte Indications			
	2	3	10	11
SET completed	80	00	00	00
Invalid command	C0	C2	00	00
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

The SET coded COMMAND macro requires two words of storage.

EXAMPLE – SET

The communications adapter must be set to an asynchronous, 600 baud, USASCII, even parity transmission line. Chaining is used.



STATUS – PLACE LINE STATUS IN STORAGE

This operation places the hardware status in the area specified by the BUFADR operand.

Name	Operation	Operands
[label]	COMMAND	OPCODE=STATUS ,BUFADR=data-name [,TIMER=integer]

NAME

An alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

OPCODE=STATUS A required entry specifying the executable command.

BUFADR= The symbolic address of the storage area containing the status word. The storage address must be aligned on a word boundary. The data-name must be alphanumeric starting with an alphabetic character and being no longer than eight alphanumeric characters.

TIMER= An optional entry; refer to Table 8-1 for integer values.

STATUS WORD BIT DESIGNATIONS

Status Word Bit Number	Description
0-5	Reserved.
6	Block check character received.
7	Block check character in error.
8	Data set ready (CC) on.
9	Clear to send (CB) on.
10	Received line signal detector (CF) on.
11	Secondary clear to send (SCB) on.
12	Ring indicator (CE) on.
13	Lost data because the ICA could not respond to the character rate of the line.
14	During asynchronous communications no stop bit was detected.
15	Reserved.

TERMINATION CONDITIONS

Description	PCB Byte Indications			
	2	3	10	11
Hardware line status returned	80	00	00	00
Invalid timer	C0	C6	00	00

EXPANSION STORAGE REQUIREMENTS

The STATUS coded COMMAND macro requires three words of storage.

COMMAND CODE SUMMARY

Table 8-3 summarizes information about COMMAND macro operations.

Table 8-3. Summary of COMMAND Macros

Command	Function	Command Word Length in Bytes	Operation Code (Hex)	Command Chain*	Switched or Non-Switched**	Synch or Asynch**
ENABLE	Establish line connection	2	01	Y	Both	Both
ANSWER	Answer-complete switched connection	2	1D	Y	Both	Both
RPTRING	Report ring indicator	2	19	N	S	Both
DISABLE	Break line connection	2	05	Y	Both	Both
BREAK	Write break	2	09	Y	Both	Asy
RCVBRK	Report break received	2	0D	N	Both	Asy
SET	Set line speed and code	4	11	Y	Both	Asy
STATUS	Return line status	6	04	N	Both	Both
JUMP	Execute command word at address specified	6	08	N	Both	Both
READA	Terminate read on any termination character	8	03	Y	Both	Both
READA1	Terminate read 1 character after any termination character	8	13	Y	Both	Both
READS	Terminate read on termination character supplied in list	12	07	Y	Both	Both
READS1	Terminate read 1 character after any termination character in list supplied	12	17	Y	Both	Both
READN	Terminate read on byte count only	8	0B	Y	Both	Both
WRITE	Write	8	02	Y	Both	Both
WRITET	Write with termination on received break	8	06	Y	Both	Asy
WRITEC	Write transparent control characters	8	0A	Y	Both	Syn
RESYN	Re-establish character framing	2	15	Y	Both	Syn
EOTSCH	Search for EOT	2	0F	Y	Both	Syn
CJE	Compare and jump if equal	12	10	Y	Both	Both
CJNE	Compare and jump if not equal	12	14	Y	Both	Both
JSR	Subroutine jump	6	0C	N	Both	Both
RTNJ	Return from subroutine	2	18	N	Both	Both

*Y identifies meaningful action and N identifies meaningless action that will be ignored if indicated.
**Choices not indicated as legal will, if specified, cause command program termination.

9. THE TERMINAL CATALOG ROUTINE

INTRODUCTION

The Terminal Catalog routine is a program under control of the operating system that the logical-level programmer must use to define terminals to the system. This routine is independent of system generation, so that terminal definitions may be changed without regenerating the system.

TERMINAL CATALOG EXECUTION

The logical-level user defines his terminals to the Terminal Catalog routine by the TERMDEV macro. This macro defines the terminal name, device type, line speed, and other pertinent information. The source-level macros are assembled and inserted into the job stream explained in the following text.

If terminals are added to the system, the user must redefine all current terminals to Terminal Catalog, in addition to defining the new ones. Likewise, if terminals are deleted, the entire set of terminal definitions must be processed by terminal catalog, minus the definitions to be dropped.

CONTROL LANGUAGE FOR TERMINAL CATALOG

The TRMDEV macros are to be coded as discussed above. The source deck is then to be inserted in the job stream exhibited below which when executed will place the terminal descriptions in $NUCLIB.

```
//JOB     NAME=JOBNAME

//EXEC    PGM=ASM

//PAR     OMEM1=TPTDT0,OMEM2=$TEMPCRD,OBJECT=YES

//DEF     ID=INPUT,DEV=SYSCRD

//DEF     ID=LIST,DEV=PRINTER

//DEF     ID=OUTPUT1,FIL=$TEMPOBJ,STA=T,DEV=DISC,NUM=500,SIZ=256,BLK=1,CSD=NO,
//        CON=YES

//DEF     ID=OUTPUT2,FIL=$TEMPCRD,STA=T,DEV=DISC,NUM=500,SIZ=80,BLK=1,
//        CSD=NO

//EXEC    PGM=$TPDIRG
```

```
//DEF     ID=OUT2,FIL=$TEMPCRD,STA=T,DEV=DISC

//DEF     ID=BLDRES,FIL=$TEMPBLD,STA=T,DEV=DISC,NUM=500,SIZ=80,BLK=1,
//        CSD=YES,CON=YES

//DEF     ID=SEGCARD,FIL=$TEMPSEG,STA=T,DEV=DISC,NUM=200,SIZ=80,BLK=1,
//        CSD=YES,CON=YES

//EXEC    PGM=LINKEDT

//PAR     PGM=TPTDT0,LST=XREF,LSD=NO,ORG='0',SRH=NO

//DEF     ID=LIST,DEV=PRINTER

//DEF     ID=INPUT,FIL=$TEMPOBJ,STA=T,DEV=DISC

//DEF     ID=DIR,FIL=$TEMPSEG,STA=T,DEV=DISC

//DEF     ID=OUTPUT,FIL=$OSRSDNTLIB†,STA=(P,0),DEV=DISC,CAT=NO,VOL=system volume†

//EXEC    PGM=$BLDRES

//PAR     DEVADDR=00*,FULLGEN=NO,STNDLOC=YES[,LISTDIR=NO]

//DEF     ID=INPUT1,FIL=$OSRSDNTLIB†,STA=(P,O),VOL=system volume†

//DEF     ID=INPUT2,FIL=$MSGLIBINPUT,STA=P,VOL=system volume †

//DEF     ID=DRECTVES,FIL=$TEMPBLD,DEV=DISC,STA=T

//DEF     ID=LIST,DEV=PRINTER

//DEF     ID=OBJLIB,FIL=$TEMPOBJ,STA=T,VOL=DISC

//DATA    FIL=SYSCRD
TPTDT0  CSECT
          .
          .
          .
        TRMDEV statements
          .
          .
          .
        END
/*
//EOJ
```

---

†The two file names used above and listed below should be changed to conform to each installation. $OSRSDNTLIB should be replaced by the name of the file which contains all the load modules — the load module library. System volume should be the volume name of the disc pack which contains $NUCLIB.

*This assumes that the catalog will be placed on the running system.

TRMDEV – TERMINAL CATALOG MACRO

This macro provides the keywords that describe terminal characteristics. One terminal may be described more than once providing each description is listed under a new logical name.

Name	Operation	Operands
terminal name	TRMDEV	DEVICE={M1240 / MRX / MRXW/O / TTYKSR / TTY / SYN} [,ID={C / X}'1-22 characters...'] ,LINECOD={ASCNO / ASCEVEN / ASCODD / EBCDIC} [,SPEED={110 / 150 / 300 / 600}] [,ENDLIST={ALL / NONE / X'xxxxxxxx'}] ,TRANSIZ=integer ,LINETYP=X'xx' ,EOU=X'xx' ,SEOU=X'xxxx' ,SEOM=X'xxxx' ,FFRAME=X'xxxx' [,NULLS=integer] [,PAD=X'xx'] [CNKTIME={0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9}] [,RCVTIME={0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9}] [,NEWLINE=X'xxxx'] [,RETRIES=integer] [,ITBSIZ=integer] [,LRC={NO / YES}] [,TAPE=X'wwxxyyyyzzzz'] [,TRNSLAT={YES / NO}]

NAME

A required entry designating the logical name of the terminal being cataloged. The logical name of the terminal must be an alphanumeric name starting with an alphabetic character and being no longer than eight alphanumeric characters.

OPERANDS

DEVICE= A required entry specifying one of the following I/O devices:

- M1240, MRX — Memorex terminal — not write only
- MRX w/o Memorex terminal — write only
- TTYKSR, TTY — Teletype 33 , 35, 37, or 38 (ASR or KSR)
- SYN — Any synchronous processor or terminal that uses the line control defined in Appendix H.

ID= An optional entry. From 1 to 22 characters are used by Logical TCOM to validate a terminal connection. When omitted, the system performs no ID validation. If the input terminal transmits in EBCDIC, the ID must be in character format. If the terminal transmits in USASCII, the ID must be in the hexadecimal format of USASCII characters.

LINECOD= An optional entry specifying the line transmission code for the terminal. The line codes are:

- ASCNO — USASCII code with no parity bit
- ASCEVEN — USASCII code with even parity
- ASCODD — USASCII code with odd parity
- EBCDIC — 8-level EBCDIC code

If the defined terminal is asynchronous and this operand is omitted, ASCEVEN is assumed. On the other hand, if the defined terminal is synchronous and the operand is omitted, EBCDIC is assumed.

SPEED= An optional entry. Specifies the bps rate of the terminal. Entries for this parameter must be 110, 150, 300, or 600. If this operand is omitted, the rate defaults to 110 if the equipment can operate at that speed and 600 if it cannot.

ENDLIST= An optional entry used to define asynchronous received termination characters. When the terminal being defined is synchronous, this operand is ignored and the ENQ, ETB, ETX, and EOT termination characters are assumed by Logical TCOM. Under asynchronous operation, the user has the option of terminating a received message when any termination character appearing in the following table is received, when the number of text characters received equals the integer specified by the TRANSIZ operand, or when any one of the one to four termination characters selected by the user from the table is received.

Termination Control Characters	EBCDIC	ASCII
EOT	37	04
ETX	03	03
ETB	26	17
CR	0D	0D
ENQ	2D	05
NAK	3D	15
CAN	18	18

If the user wishes to terminate a message on the receipt of any termination character, he selects the ALL option or omits the ENDLIST operand. If the user wishes to terminate the input message when the number of received text characters equals the integer specified by the TRANSIZ operand, he selects the NONE operand. When the user specifies any one of one to four termination characters to terminate a received message, he selects the X'xxxxxxxx' option. The X'xxxxxxxx' option allows the user to list the hexadecimal representation of up to four termination characters. If fewer than four termination characters are selected, they must be left-justified in the field with the remaining positions filled with nulls.

TRANSIZ= A required entry specifying the longest transmission that can be received by the computer. This integer must be large enough to include data, and control and null characters appearing in RECEIVE macro's WKAREA field but not so large as to consume more storage than available.

LINETYP= A required entry from the Modem/Line Type column in Table 2-2. The line type entry describes the hardware line connection for the terminal. Placement of one control character may be right- or left-justified in the field.

EOU= A required entry for asynchronous lines specifying the line code, in hexadecimal, of the termination character for received TRANSUNITS. The character specified must be one of the characters designated by the ENDLIST operand. This operand may be omitted when describing a synchronous terminal since Logical TCOM assumes an ETB character for TRANSUNIT termination.

SEOU= A required entry for asynchronous terminals representing a maximum of two termination characters used to terminate a computer to asynchronous terminal TRANSUNIT. All control characters must appear in the proper line code in hexadecimal in the field. If only one character is used, the remaining character position should be filled with X'FF'. Placement of one control character may be right- or left-justified in the field. This operand may be omitted when describing a synchronous terminal, since Logical TCOM assumes an ETB character for TRANSUNIT termination.

SEOM= A required entry for asynchronous terminals representing a maximum of two characters that will be used to terminate a message transmission from a terminal. All control characters must appear in the proper line code in hexadecimal in the field. If only one character is used, the remaining character position should be filled with X'FF'. Placement of one control character may be right- or left-justified in the field. This operand may be omitted when describing synchronous terminals, since Logical TCOM assumes an ETX character for MESSAGE termination.

FFRAME= A required entry for asynchronous lines representing a maximum of two control characters used to precede any transmission to the terminal. All control characters must appear in the proper line code in hexadecimal in the field. If only one character is used, the remaining character position should be filled with X'FF'. Placement of one control character may be right- or left-justified in the field. This operand is omitted for synchronous data transmission.

NULLS= An optional integer value ranging from 1 to 99 specifying the number of null characters to be inserted in the message at transmission time. TTY terminals require *one* null to follow the NEWLINE characters in a transmission. MRX terminals have individual requirements for the minimum number of characters in a transmission.

MRX Terminal	Characters Required	
	96-Character Belt	48-Character Belt
10 cps	6	3
15 cps	9	5
30 cps	20	10
60 cps	41	21
120 cps	0	0

The default is 1 for TTY's, or a TCOM calculated character count for the terminal based on a 96-character belt.

This operand is omitted if the terminal is synchronous, or the asynchronous terminal does not require null characters.

PAD= An optional entry specifying the pad character's hexadecimal representation. This entry must be omitted when using asynchronous terminals.

CNKTIME= An optional entry specifying the timer to be used in waiting for the terminal to connect to. The keyword entries range from 0 to 9. The time periods range from 1 second for integer one to 1800 seconds for integer nine. (See the integer definition chart below.) If integer 0 is used, no time limit is placed on the connection time. When this operand is omitted, and a switched answer terminal is being connected, no time limit is placed on the connection time.

Integer	Time Limit
0	No timing
1	1 second
2	3 seconds
3	15 seconds
4	30 seconds
5	60 seconds
6	180 seconds
7	300 seconds
8	600 seconds
9	1800 seconds

RCVTIME= An optional entry specifying the maximum period logical communications allows any read service request to exist without being completed. The integer values represent the same time periods specified in the CNKTIME operand. If this option is omitted and a synchronous terminal is referenced, the time period defaults to 3 seconds. Likewise, if an asynchronous terminal is referenced, the time period defaults to 3 minutes.

NEWLINE= An optional entry that defines the control characters that will return the carriage and advance the paper one line. All control characters must appear in the proper line code in hexadecimal in the field. If only one character is needed, the remaining character position should be filled with X'FF'.

RETRIES= An optional entry defining the number of times a transmission will be repeated if a transmission error occurs. The number of retries ranges from 0 to 15. If this operand is omitted, the number of retries defaults to 6.

ITBSIZ= An optional entry specifying the size of each intermediate block or message segment containing ITB. This entry is required if ITB is to be used in transmitting to and from this terminal.

LRC= An optional entry specifying that an LRC check is to be made for all transmitted data.

TAPE= A required parameter for terminals with auxiliary tape devices. The entry must be coded TAPE=wwxxyyyyzzzz.

Where:

ww	is the read initiate character
xx	is the read continue character
yyyy	is the write initiate character
zzzz	is the stop write character

None of the TAPE entries can be omitted. If not applicable to a particular terminal, they must be coded as 00.

For teletypes that will operate in an unattended mode, the following TAPE= entries are recommended:

ww	=	X'11' (DC1-Reader on)
xx	=	X'FF' (Fill)
yyyy	=	X'1412' (DC4-Punch off, DC2-Punch on)
zzzz	=	X'1314' (DC3-Reader off, DC4-Punch off)

TRNSLAT= An optional entry that specifies whether a message is to be translated (TRNSLAT=YES) or untranslated (TRNSLAT=NO). The default is YES.

10. OPERATOR CONSOLE COMMANDS AND MESSAGES

The operator can use the operator console STATUS command to request the status for individual lines, individual terminals, all terminals, and/or all terminals and lines.

FORMAT OF STATUS COMMAND

The STATUS command format is:

$$\text{T STATUS} \left\{ \text{ALL, ALLL, ALLT, n, terminal name} \right\}$$

One or more of the parameters can be specified in any sequence. They must be separated by commas and the command must be $\leq$ 51 characters in length.

PARAMETERS

The STATUS command parameters are:

Parameter	Description
ALL	ALL requests the status of all lines and of those terminals being used by logical TCOM.
ALLL	ALLL requests the status of all lines.
ALLT	ALLT requests the status of those terminals being used by logical TCOM.
n	n is 1-9 or A-F and requests the status of the line specified by n.
terminal name	This parameter requests the status of the terminal name specified. This terminal name must not be ALL, ALLL, ALLT, or A-F. To obtain the status of a terminal with one of these names, the parameter ALLT or ALL must be used.

TCOM CONSOLE RESPONSES TO OPERATOR STATUS REQUESTS

TCOM responds to a status request with one or more of the following types of message:

- Input parameter error message
- Line status message
- Terminal status message

INPUT PARAMETER ERROR MESSAGE

STATUS command parameter error messages have the format:

```
## I TPST0018 X. . .X
```

where:

##	indicates that no partition is assigned to the status program.
I	indicates an informative console message that does not require a response.
TP	indicates a telecommunications message.
ST	indicates a TCOM status message.
001	indicates a parameter error in the STATUS command.
8	indicates that the parameter in error was not processed.
X. . .X	is replaced by a list of the parameters in error.

LINE STATUS MESSAGES

Line status messages issued in response to an operator STATUS command are in the format:

```
## I TPST0000 LNE=aa, {UNASSIGNED
                      {TYPE=bb,PID=c,ENBL={Y},CMD=ee,
                                          {N}
TERM=terminal name}
```

where:

##	indicates that no partition is assigned to the status program.
I	indicates an informative console message that does not require a response.
TP	indicates a telecommunications message.
ST	indicates a TCOM status message.
000	indicates that the message contains status information.
0	indicates an informative rather than an error message.
LINE=aa	aa is the physical line address.
UNASSIGNED	indicates that the line is not assigned to a partition.
TYPE=bb	bb is the equipment type of the hardware. (See Table 2-1 for modem/line types supported by TCOM.
PID=c	c is the partition to which this line is assigned.
ENBL={Y / N}	Y indicates that this line is enabled. N indicates that this line is not enabled.
CMD=ee	ee represents the op code of the command being executed at the time of the STATUS request. ee is 00 if no command is in progress.
TERM=terminal name	terminal name is the terminal associated with the line if the line is physically enabled. This part of the message is included only for logical TCOM.

TERMINAL STATUS MESSAGES

Terminal status messages issued in response to an operator STATUS command are in the format:

```
## I TPST0000 {LOG TCOMM IDLE                                    }
              {TERM=terminal name, EQP TYPE=h,{LOG ENBL}          }
                                              {PHY ENBL}          },
                                              {NOT ENBL}          }
PID=i,LNE=aa,CMD=ee}
```

where:

##	indicates that no partition is assigned to the status program.
I	indicates an informative console message that does not require a response.
TP	indicates a telecommunications message.
ST	indicates a TCOM status message.
000	indicates that the message contains TCOM status information.
0	indicates an informative message rather than an error message.
LOG TCOMM IDLE	indicates that logical TCOM is inactive. No messages concerning terminal status can be sent to the console.
TERM=terminal name	terminal name is the name of the logical TCOM terminal.
EQP TYPE=h	h is the type of terminal.

h	Terminal
1	Memorex terminal, not write only
2	TTY 33, 35, 37, 38, ASR or KSR
3	Synchronous terminal
5	Memorex terminal, write only

LOG ENBL	indicates that the terminal is logically enabled.
PHY ENBL	indicates that the terminal is physically enabled.
NOT ENBL	indicates that the terminal is not enabled.
PID=i	i specifies the partition that this terminal is associated with.
LNE=aa	aa specifies this terminal's physical line address. This part of the message is included only if the terminal is physically enabled.
CMD=ee	specifies the op code of the command being executed at the time of the STATUS request. ee is 00 if no command is in progress. This part of the message is included only if the terminal is physically enabled.

11. TCOM CONTROL LANGUAGE STATEMENTS

INTRODUCTION

Control Language statements communicate information about the applications program to the system. The statements initiate the job, control the sequential execution of programs, monitor the progress of the jobs, and assign communications lines needed to complete each program. This section only covers those statements that assign communications lines and links them to terminals and logical communications buffers. This section does, however, show two examples of job control sequences. The first example demonstrates the statements used for a logical communications operation, and the second example demonstrates the statements used for a physical communications operation. Complete information on job control can be found in the **MRX/OS Control Language Services, Basic Reference** manual.

LOGICAL COMMUNICATIONS CONTROL STATEMENTS

All operands defined below apply to the Control Language //DEFINE statement. The function of these operands is to link communications lines to terminals, assign terminals to the applications program, and define the storage requirements for the number of buffers needed for each terminal.

{IDENT / ID} =terminal name

This operand specifies the terminal to be assigned to the applications programs. The name used must be derived from the name field of the Terminal Catalog TRMDEV macro.

{DEVICE / DEV} = {(00i,. . .,00n) / TPyy,m}

This operand links one terminal to one or more communications lines. The "00i,. . .,00n" option assigns a terminal to a maximum of eight lines. If more than eight lines are linked to a terminal, two DEVICE statements must be used. The "i" to "n" hexadecimal characters can range from 1 for the first line to F for the fifteenth line. If the MRX/40 system is used, the hexadecimal character cannot exceed 7 and only seven lines can be linked to one terminal. The "TPyy,m" option allows the operating system to link the terminal to one or more lines. The "yy" integer values are selected from the Modem Line Type column in Table 2-1, Modem/Line Type Descriptions. The Modem Line Type column describes the line equipment needed to support the terminal. The integer "m" defines the number of TPyy lines that will be linked to the terminal. The "m" integer should never exceed the value 8 if the MRX/50 system is used, seven if the MRX/40 system is used, or the number of particular line types attached to the system. Only one non-switched line can be linked to a terminal. Switched calling terminals may be linked to more than one line by using the "00i, . . . ,00n" option or the "TPyy,m" option. Switched answer lines should only be linked to one or more lines by using the "00i, . . . ,00n" option.

When assigning terminals to lines the following rules should be observed unless the particular application dictates otherwise.

1. All terminals should operate at the same speed and with the same line code.

2. Synchronous and asynchronous devices must never be assigned to the same line.

3. All terminals assigned to one line should either have unique identification numbers or no identification numbers at all.

BUFFER=(number,size)

A required operand. Number and size are two integer values that specify the number of buffers of a given size the user would like to pool. Since the buffers are pooled, the BUFFER operand does not have to refer to the defined terminal. All buffers may be defined in one //DEFINE statement using multiple BUFFER operands or be interspersed with any or all of the terminal definition keywords.

FIL=TP

A required operand.

EXAMPLE – LOGICAL COMMUNICATIONS CONTROL STATEMENTS

Figure 11-1 shows the control statements for a job called LIOEX. The job contains one program called EX1 that transfers data to two remote terminals called TSRCV and TLRCV from a disc file called DSTR4. The TSRCV terminal is attached to a switched-calling manual-dial line and the TLRCV terminal is attached to a leased line.

```
//JOB NAME=LIOEX,USER=ID005,PRIORITY=9,TYPE=1
//EXECUTE PGM=EX1
//DEFINE IDENT=TRSCV,DEVICE=(TP86,1),FILE=TP
//DEFINE IDENT=TLRCV,DEVICE=007,FILE=TP
//DEFINE BUFFER=(2,135)
//DEFINE IDENT=DSTR4,FILENAME=TOATA,STATUS=(P,I),VOL=18AGB2
                        .
                        .
                        .
                        .
                        .
                (Application Program)
                        .
                        .
                        .
                        .
                        .
//EOJ
```

Figure 11-1. Logical Communications Control Statements — Example

PHYSICAL COMMUNICATIONS CONTROL STATEMENTS

The physical communication Control Language statements assign communications lines to the user's program. It is the responsibility of the user to control the terminals in his applications program. All operands apply to the //DEFINE statement.

IDENT=line name

A required entry that specifies the line name to be assigned to the applications program. This name must correspond to the IDENT entry in the operand field of the OPEN, CLOSE, PCB and ABORT macros. The line name must start with an alphabetic character and be no longer than eight alphanumeric characters. Line names shorter than eight characters are left-justified in the field with trailing blanks.

DEVICE $\left\{ \begin{matrix} =(00i,\ldots,00n) \\ =TPyy,m \end{matrix} \right\}$
DEV

This required operand assigns one line-name to one or more communications lines. The "00i,. . .,00n" option assigns one terminal to n lines where n cannot exceed 8. The "i" to "n" hexadecimal character can range from 1 for the first line to F for the fifteenth line. If the MRX/40 system is used, the hexadecimal character cannot exceed 7 and only seven lines can be assigned to one line-name. The "TPyy,m" option allows the operating system to

assign the line-name to one or more lines. The "yy" integer values are selected from the Modem Line Type column in Table 2-1. The Modem Line Type column describes the line equipment needed to support the terminal. The integer "m" defines the number of TPyy lines that will be assigned to the line-name. The "m" integer should never exceed the value of 8 if the MRX/50 system is used, 7 if the MRX/40 system is used, or the number of particular line types attached to the system. Only one non-switched line can be assigned to a line-name. A switched-calling line-name may be assigned to more than one line by using the "00i, . . . ,00n" option or the "TPyy,m" option. Switched answer lines should only be assigned to one or more line-names by using the "00i, . . . ,00n" option.

When assigning terminals to lines the following rules should be observed unless the particular application dictates otherwise.

1. All terminals should operate at the same speed and with the same line code.

2. Synchronous and asynchronous devices must never be assigned to the same line.

3. All terminals assigned to one line should either have unique identification numbers or no identification numbers at all.

FILE=PIO

A required operand.

EXAMPLE – PHYSICAL COMMUNICATIONS CONTROL STATEMENTS

Figure 11-2 shows the control statements for a job called PIOEX. The job contains one program called EX2 that transfers data to a remote terminal on a line called TRCV from a disc file called DSTR3.

```
//JOB NAME=PIOEX,USER=ID005,PRIORITY=8,TYPE=1
//EXECUTE PGM=EX2
//DEFINE IDENT=TRCV,DEVICE=009,FILE=PIO
//DEFINE IDENT=DSTR3,FILENAME=PTDATA,STATUS=(P,I),VOL=18AGB2
                        .
                        .
                        .
                        .
                        .
                (Application Program)
                        .
                        .
                        .
                        .
                        .
//EOJ
```

Figure 11-2. Physical Communications Control Statements — Example

A. EBCDIC AND ASCII CHARACTER ASSIGNMENTS

Table A-1. EBCDIC Character Set

EBCDIC Character	Hex Code
NUL	00
SOH	01
STX	02
ETX	03
PF	04
HT	05
LC	06
DEL	07
SMM	0A
VT	0B
FF	0C
CR	0D
SO	0E
SI	0F
DLE	10
DC1	11
DC2	12
DC3	13
RES	14
NL	15
BS	16
IL	17
CAN	18
EM	19
CC	1A
IFS	1C
IGS	1D
IRS	1E
ITB (IUS)	1F
DS	20
SOS	21
FS	22
BYP	24
LF	25
EOB/ETB	26
ESC/PRE	27
SM	2A
ENQ	2D
ACK	2E
BEL	2F
SYN	32
PN	34
RS	35
UC	36
EOT	37
DC4	3C
NAK	3D
SUB	3F
SP	40

EBCDIC Character	Hex Code
¢	4A
.	4B
<	4C
(	4D
+	4E
|	4F
&	50
!	5A
$	5B
*	5C
)	5D
;	5E
¬	5F
-	60
/	61
'	6B
%	6C
_	6D
>	6E
?	6F
:	7A
#	7B
@	7C
'	7D
=	7E
"	7F
a	81
b	82
c	83
d	84
e	85
f	86
g	87
h	88
i	89
j	91
k	92
l	93
m	94
n	95
o	96
p	97
q	98
r	99
s	A2
t	A3
u	A4
v	A5
w	A6

Table A-1. EBCDIC Character Set (Continued)

EBCDIC Character	Hex Code
x	A7
y	A8
z	A9
A	C1
B	C2
C	C3
D	C4
E	C5
F	C6
G	C7
H	C8
I	C9
J	D1
K	D2
L	D3
M	D4
N	D5
O	D6
P	D7
Q	D8

EBCDIC Character	Hex Code
R	D9
S	E2
T	E3
U	E4
V	E5
W	E6
X	E7
Y	E8
Z	E9
0	F0
1	F1
2	F2
3	F3
4	F4
5	F5
6	F6
7	F7
8	F8
9	F9

Table A-2. ASCII Character Set

ASCII Character	Hex Code
NUL	00
SOH	01
STX	02
ETX	03
EOT	04
ENQ	05
ACK	06
BEL	07
BS	08
HT	09
LF	0A
VT	0B
FF	0C
CR	0D
SO	0E
SI	0F
DLE	10
DC1	11
DC2	12
DC3	13
DC4	14
NAK	15
SYN	16
ETB	17
CAN	18
EM	19
SUB	1A
ESC	1B
FS	1C
GS	1D
RS	1E
ITB(US)	1F
SP	20
!	21
"	22
#	23
$	24
%	25
&	26
'	27
(	28
)	29
*	2A
+	2B
,	2C
-	2D
.	2E
/	2F
0	30
1	31

ASCII Character	Hex Code
2	32
3	33
4	34
5	35
6	36
7	37
8	38
9	39
:	3A
;	3B
<	3C
=	3D
>	3E
?	3F
@	40
A	41
B	42
C	43
D	44
E	45
F	46
G	47
H	48
I	49
J	4A
K	4B
L	4C
M	4D
N	4E
O	4F
P	50
Q	51
R	52
S	53
T	54
U	55
V	56
W	57
X	58
Y	59
Z	5A
[	5B
\	5C
]	5D
⌜ ⌝	5E
_	5F
`	60
a	61
b	62
c	63
d	64

Table A-2. ASCII Character Set (Continued)

ASCII Character	Hex Code
d	64
e	65
f	66
g	67
h	68
o	69
j	6A
k	6B
l	6C
m	6D
n	6E
o	6F
p	70
q	71

ASCII Character	Hex Code
r	72
s	73
t	74
u	75
v	76
w	77
x	78
y	79
z	7A
{	7B
\	7C
}	7D
~	7E
DEL	7F

B. PHYSICAL AND LOGICAL TCOM STORAGE REQUIREMENTS

PHYSICAL TCOM STORAGE REQUIREMENTS

The storage requirements for Physical TCOM are outlined below. The storage used by Physical TCOM is added to the storage required by other system programs to determine the total storage required by the operating system.

1. Basic Telecommunications Capability

   a. Physical Communications and Driver Functions (including operator console interface) contains: 2046

2. Physical Communications and Driver Routine Options

   The table below indicates the increase in storage required to support each line type individually. Combinations of line types are not necessarily additive; the maximum Physical TCOM storage requirement is 3928.

Line Type	Storage Increase
80	134
84	308
85	456
86	396
88	308
8A	396
8C	456
8E	544
98	360
9A	448
A4	804
A5	804
A6	892
A8	660
AA	748
B4	934
B5	934
B6	1022
B8	790
BA	878

3. Physical Communication Tables – Terminal/Line Dependent

   a. Line Parameter Table (LPT) – Half/Duplex with Timer and Dial: 80

   b. Line Address Table (LAT) – One table required for system physical communications: 32

LOGICAL TCOM STORAGE REQUIREMENTS

Logical communications requires a total of 1586 bytes. The first 768 bytes are for the root module, and the last 818 are for overlays.

The formula below can be used to calculate the storage requirements for Logical TCOM. The storage required is comprised of tables and buffers which reside in the partition containing the user's program. If the user wishes to calculate the amount of storage consumed by the operating system, including Logical TCOM, he must total the storage required for the operating system, Physical TCOM, and Logical TCOM. (Since Logical TCOM uses Physical TCOM to complete the communications operation, the total storage required by Logical TCOM must include the storage required by Physical TCOM.)

$$I = 34+72T+80L+\sum_{i=1}^{m}[(S_i+14)N_i+8]$$

Where:

- I = The number of bytes of storage consumed.
- T = The number of terminals named in the IDENT operands of the MRX/OS Control Language //DEFINE statement.
- L = The number of lines named. This is calculated by totaling the number of Control Language //DEVICE statement operands.
- i = An index value ranging from one to "m".
- m = The number of unique BUFFER operands defined by the Control Language //DEFINE statement.
- S_i = The size of the buffers in the i^{th} BUFFER keyword defined by MRX/OS Control Language.
- N_i = The number of buffers of size S_i defined by Control Language.

C. PHYSICAL COMMAND BLOCK

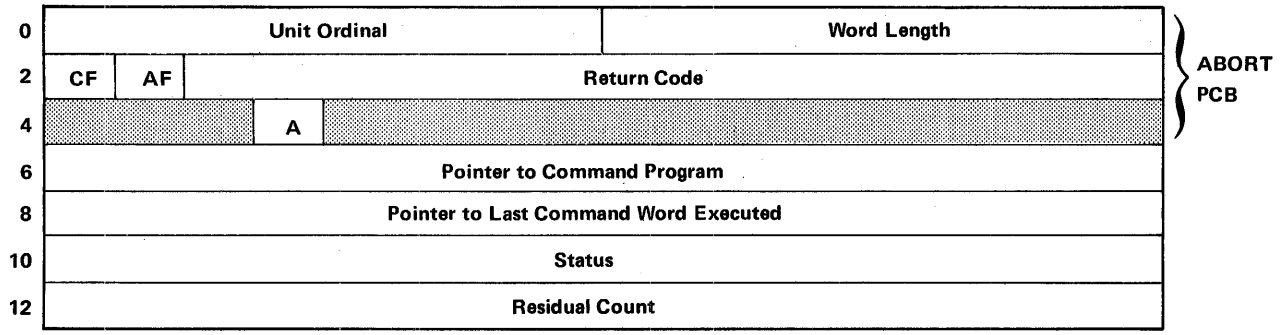


Figure C-1. Physical Command Block Structure

The fields of the PCB are defined in the following table.

Displacement		Name	Description
Byte	Bits		
0	0-7		Unit ordinal.
	8-15	Word Length	Specifies the word length of the physical command block. Physical macros have a word length of seven except for the ABORT macro which has a word length of three.
2	0	Complete Flag	Set by the system when the I/O request is complete.
	1	Abnormal Flag	Set by the system indicating abnormal completion of the I/O request. When set, an abnormal software condition exists.
	2-15	Return Code	Set by the system indicating the termination conditions. For the bit designations, refer to Table C-1, PCB Return Codes.
4	0-2		Reserved.
	3	Abort Request	Set when the ABORT macro is used.
	4-15		Reserved.
6	0-15	Pointer to Command Program	Set by the CPADR operand in the PCB macro or loaded by the CP operand in the EXCP macro.
8	0-15	Pointer to Last Command Word Executed	Set by the driver upon completion of the I/O request. This block indicates the address of the last command word executed.
10	0-15	Status	Set by the TCOM system at the completion of the I/O request. For the bit designations refer to Table C-2, PCB Status.
12	0-15	Residual Count	Set by TCOM when the I/O request terminates. The hexadecimal count indicates the number of bytes remaining in the buffer.

Table C-1. PCB Return Codes

Complete, Abnormal and Return Code Field	Meaning	Description	Comments
X'8000'	Normal Completion	The command program was executed to completion.	Status field should be checked for additional information.
X'C030'	Invalid Unit Ordinal	The specified unit ordinal is not in the system.	Status field will not be updated for this request.
X'C0C2'	Invalid Command	The command addressed in the last command word executed field of the PCB is invalid for either of the reasons given in the next column.	1. A data transfer command was issued to a line which was not enabled. 2. This command was issued to a line which is incapable of executing the command. 3. A control command was issued to a line at an inappropriate time (that is, an ENABLE command was issued to an enabled line).
X'C0C4'	Zero Byte Count	The data transfer command contained a zero count field.	A request to transfer zero bytes is invalid.
X'C0C6'	Invalid Timer	The timer specified in the last command word executed field of the PCB is invalid.	The timer code specified has not yet been assigned.

Table C-2. PCB Status

Status	Meaning	Description	Comments
X'8000'	Abort Requested	The command program was aborted via the user's request.	The command in execution is stopped and the communication request is aborted.
X'4000'	Timer Expired	The timer requested with this command expired.	The command in execution is stopped and the communication request is terminated.
X'2000'	Termination Character Received	The communication driver detected a requested termination character in the input data.	The command and the command program are terminated.
X'1000'	BREAK Received	A break was detected on the addressed asynchronous line.	*If the command in execution is a WRITET, execution will stop immediately. The execution of any other command will not be altered. This only applies to asynchronous lines.
X'0800'	Intermediate BCC Detected	A BCC status was received and the input buffer was not terminated.	*The command terminated when the termination character was received. Input buffer contains ESB's.
X'0400'	Software Lost Data	One or more input characters were received by the communication driver and no read command for this line was pending.	*The output data transfer command in execution (for 103A only) or the next data transfer command is executed to completion. The command program is then terminated.
X'0200'	Transmission Error	No stop-bit, VRC error, or a BCC/LRC error occurred on the processing unit receipt of data.	The input command is executed to completion and the command program is terminated.
X'0100'	Reserved		
X'0080'	Modem Error	A modem error was detected on the addressed line.	The command in execution and the command program is terminated upon detection of this condition. The user can issue a RECEIVE with STATUS=YES for more information.
X'0040'	Line Disconnect Detected	A line disconnect was detected on the addressed line.	The command in execution and the command program are terminated upon detection of this condition.
X'0020'	Reserved		
X'0010'	Reserved		
X'0008'	Ring Indicator	The ring indicator was received for the addressed line.	This status bit is set only when the RPTRING command is pending. The command program is terminated upon detection of this condition.
X'0004'	Hardware Lost Data	A Processing Unit overload was detected and the addressed line adapter was unable to enter the assembled character into the common control (hardware) queue.	*The command in execution or the next command issued is executed to completion and the command program is then terminated.
X'0002'	Hardware Timer Expired	The SYN-3 second hardware timer expired.	*The command in execution or the next command to be executed will continue to completion. The command program will then be terminated.
X'0001'	Reserved		

*These conditions can occur simultaneously with each other. Also they may occur with only one of those conditions not noted.

D. COMMAND WORD EXPANSIONS

Five command word expansions are used for the COMMAND macros. In the following material each command is associated with the proper format, and the function of each bit in the format is discussed.

READA, READA1, READN, WRITE, WRITET, AND WRITEC EXPANSION

Figure D-1 shows the command word expansion for these COMMAND macros.

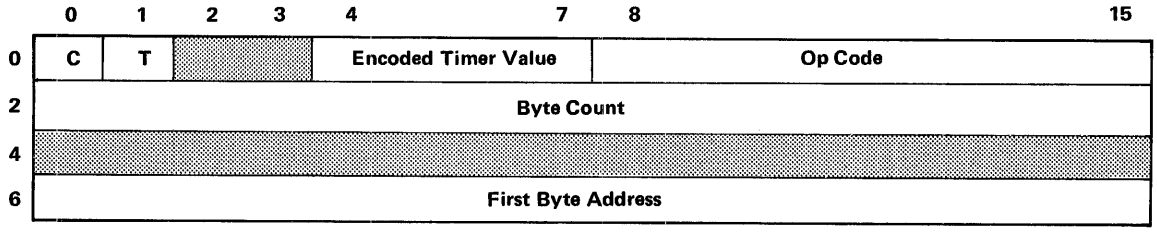


Figure D-1. Command Word Expansion One

The fields of this expansion are defined in the following table.

Displacement		Name	Description
Byte	Bits		
0	0	Chain	Set by inclusion of the CHAIN=YES operand. The CHAIN operand indicates that another command word follows.
	1	Timer	Set by inclusion of the TIMER=integer operand.
	2-3		Reserved.
	4-7	Encoded Timer Value	If the timer is used, this field specifies the binary value used to represent the elapse time. The Time Limit, Table 8-1, defines the integer values for each time limit.
	8-15	Op Code	The binary equivalent of the hexadecimal delination of the operation codes specified for this format. The hexadecimal delineations for the COMMAND macros pertaining to this format are: 03_{16} READA, 13_{16} READA1, $0B_{16}$ READN, 02_{16} WRITE, $0A_{16}$ WRITEC, and 06_{16} WRITET.
2	0-15	Byte Count	The binary equivalent for the hexadecimal number of bytes to be transferred during the operation.
4	0-15		Reserved.
6	0-15	First Byte Address	The binary equivalent of the hexadecimal memory address pointing to the location of the first byte.

READS AND READS1 EXPANSION

Figure D-2 shows the command word expansion for the READS and READS1 coded COMMAND macros.

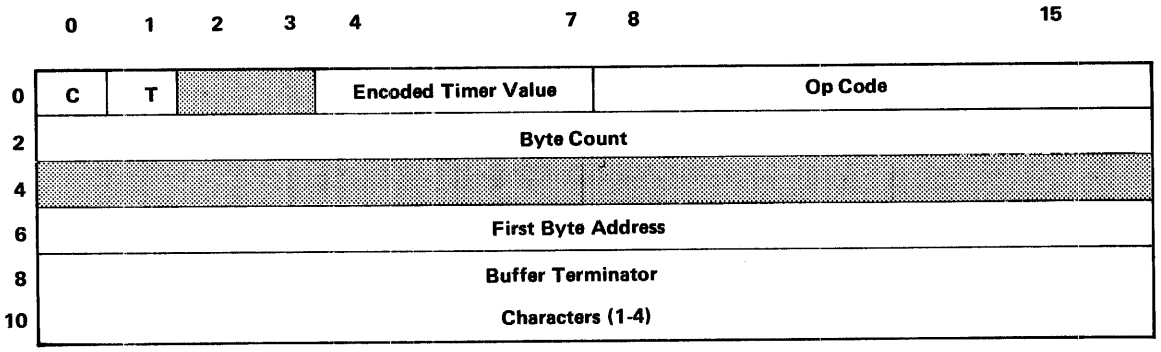


Figure D-2. Command Word Expansion Two

The fields for this expansion are defined in the following table.

Displacement		Name	Description
Byte	Bits		
0	0	Chain	Set by the inclusion of the CHAIN=YES operand indicating that another command word follows.
	1	Timer	Set by the inclusion of the TIMER=integer operand.
	2-3		Reserved.
	4-7	Encoded Timer Value	If the timer is used, this field specifies the binary numeric value used to represent the permissible elapse time. Table 8-1 defines the integer value for each time limit.
	8-15	Op Code	The binary equivalent of the hexadecimal READS operation code 07_{16} or the READS1 operation code 17_{16}.
2	0-15	Byte Count	The binary equivalent for the hexadecimal number of bytes to be transferred during the operation.
4	0-15		Reserved.
6	0-15	First Byte Address	The binary equivalent of the hexadecimal memory address pointing to the first operable byte.
8	0-15	Buffer Terminator Characters	The binary equivalent of up to four termination control characters left-justified in the field and selected from the list of termination control characters appearing in Table 8-2. They must be specified in line code.
10	0-15		

SET EXPANSION

Figure D-3 shows the command word expansion for the SET COMMAND macro.

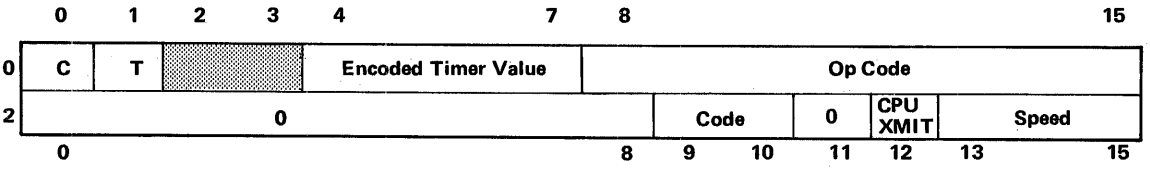


Figure D-3. Command Word Expansion Three

The fields of this expansion are defined in the following table.

Displacement		Name	Description
Byte	Bits		
0	0	Chain	Set by the inclusion of the CHAIN=YES indicating that another command word follows.
	1	Timer	Set by the inclusion of the TIMER=integer operand.
	2-3		Reserved.
	4-7	Encoded Timer Value	If the timer is used, this field specifies the binary numeric value used to represent the elapsed time. Table 8-1 defines the integer value for each time limit.
	8-15	Op Code	The binary equivalent of the hexadecimal SET operation code 11_{16}.
2	0-8		Set to zero.
	9-10	Code	A two-bit configuration specifying the line code of the enabled line. The representative bit configurations are: 00_2 for USASCII with even parity, 01_2 for USASCII with odd parity, 10_2 for USASCII without parity, and 11_2 for EBCDIC.
	11		Set to zero.
	12		Set to indicate that the CPU will transmit on the primary channel.
	13-15	Speed	A three-bit configuration specifying the baud rate of the enabled line. The representative bit configurations are: 011_2 for 1200 baud, 100_2 for 600 baud, 101_2 for 300 baud, 110_2 for 150 baud, and 111_2 for 110 baud.

RPTRING, ANSWER, BREAK, RCVBRK, ENABLE, DISABLE, RTNJ, EOTSCH, AND RESYN EXPANSION

Figure D-4 shows the command word expansion for these COMMAND macros.

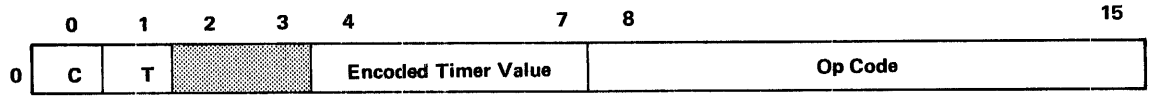


Figure D-4. Command Word Expansion Four

The fields of this expansion are defined in the following table.

Displacement		Name	Description
Byte	Bits		
0	0	Chain	Set by inclusion of the operand CHAIN=YES indicating that another command word follows. RPTRING and RCVBRK must not be chained to a successor command.
	1	Timer	Set by the inclusion of the TIMER=integer operand.
	2-3		Reserved.
	4-7	Encoded Timer Value	If the timer is used, this field specifies the binary value used to represent the time limit. Table 8-1 shows the integer value for each time limit.
	8-15	Op Code	The binary equivalent of the hexadecimal delineation of the operation codes specified for the format. The hexadecimal delimeations for the COMMAND macros pertaining to this format are: 19_{16} RPTRING, 10_{16} ANSWER, 09_{16} RCVBRK, 01_{16} ENABLE, 05_{16} DISABLE, 18_{16} RTNJ, $0F_{16}$ EOTSCH, and 15_{16} RESYN.

STATUS, JUMP, AND JSR EXPANSION

Figure D-5 shows the command word expansion for the STATUS, JUMP and JSR COMMAND macros.

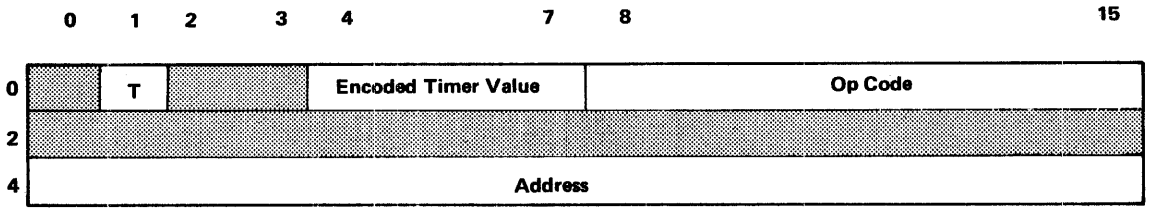


Figure D-5. Command Word Expansion Five

The fields of this expansion are defined in the following table.

Displacement		Name	Description
Byte	Bits		
0	0		Reserved.
	1	Timer	Set by inclusion of the TIMER=integer operand.
	2-3		Reserved.
	4-7	Encoded Timer Value	If the timer is used, this field specifies the binary numeric value to represent the time limit. Table 8-1 shows the integer value for each time limit.
	8-15	Op Code	The binary equivalent of the hexadecimal STATUS operation code (11_{16}) JUMP operation code (08_{16}), or JSR operation code ($0C_{16}$).
2	0-15		Reserved.
4	0-15	Address	The binary equivalent of the hexadecimal memory address pointing to the first storage location of the status field or branching address.

CJE AND CJNE EXPANSION

Figure D-6 shows the command word expansion for the CJE and CJNE COMMAND macros.

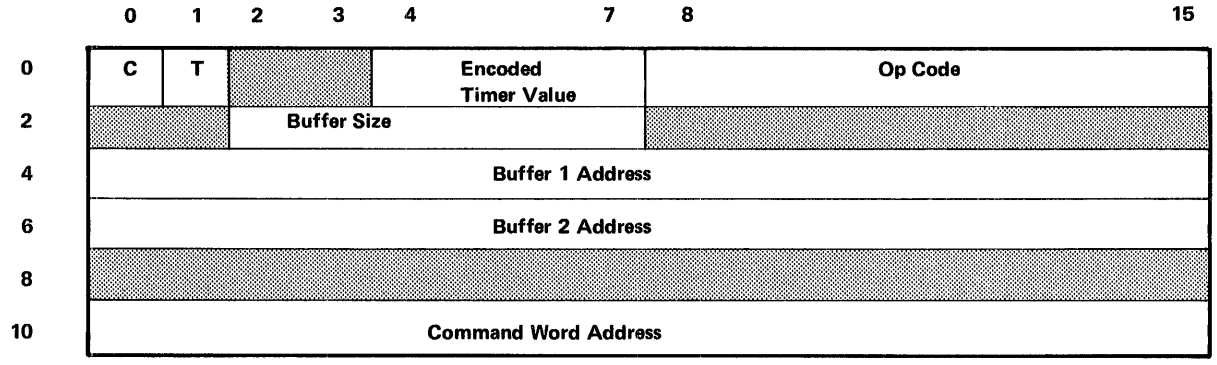


Figure D-6. Command Word Expansion 6

The fields of this expansion are defined in the following table.

Displacement		Name	Description
Byte	Bits		
0	0	Chain	Set by the inclusion of CHAIN=YES indicating that another command word follows.
	1	Timer	Set by the inclusion of the TIMER=integer operand.
	2-3		Reserved.
	4-7	Encoded Timer Value	If the timer is used, this field specifies the binary numeric value used to represent the elapsed time. Table 8-1 defines the integer value for each time limit.
	8-15	Op Code	The binary equivalent of the hexadecimal CJE operation code (10_{16}) or CJNE operation code (14_{16}).
2	0-1		Reserved.
	2-7	Buffer Size	The number of bytes in each buffer to be compared.
	8-15		Reserved.
4	0-15	Buffer 1 Address	The address of the beginning of the first field to be compared.
6	0-15	Buffer 2 Address	The address of the beginning of the second field to be compared.
8	0-15		Reserved.
10	0-15	Command Word Address	The address of the next command to be executed if the jump conditions are met.

E. USER TERMINAL CONTROL BLOCK

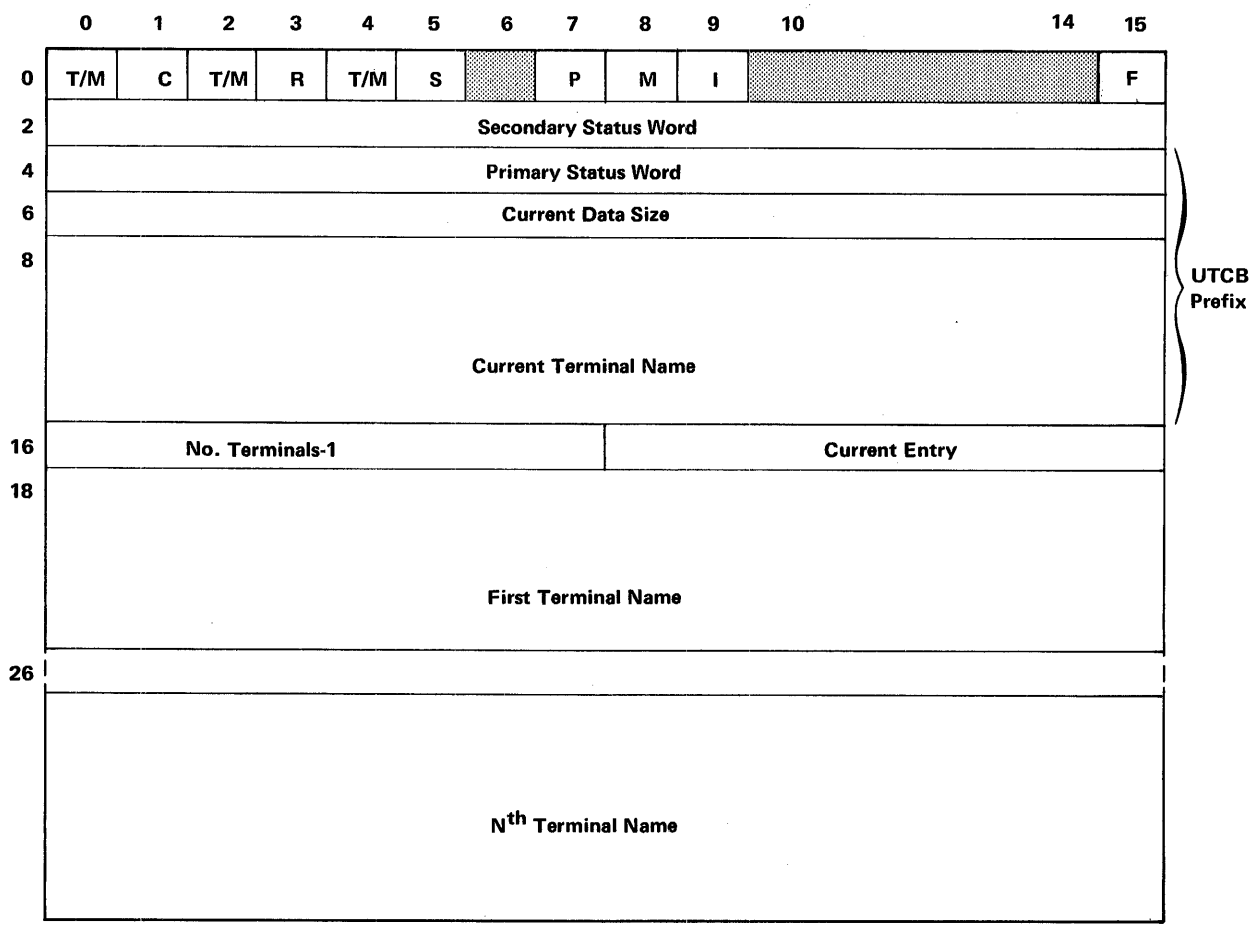


Figure E-1. User Terminal Control Block

The fields of the UTCB are defined in the following table.

Displacement		Name	Description
Byte	Bits		
0	0	TRANSUNIT/ MESSAGE	This bit is zero when in the CONVERS mode and sending by TRANSUNIT. The bit is 1 when in the CONVERS mode and sending by MESSAGE.
	1	CONVERS	Set when in the CONVERS mode.
	2	TRANSUNIT/ MESSAGE	This bit is zero when in the RECEIVE mode and sending by TRANSUNIT. The bit is one when in the RECEIVE mode and sending by MESSAGE.
	3	RECEIVE	Sets when in the RECEIVE only mode.
	4	TRANSUNIT/ MESSAGE	This bit is zero when in the SEND mode and sending by TRANSUNIT. The bit is one when in the SEND mode and sending by MESSAGE.

Displacement		Name	Description
Byte	Bits		
	5		Set when in the SEND only mode.
	6		Reserved.
	7	Prompt	Set when the terminal is to be prompted for read operations (asynchronous only).
	8		Set if terminal is master synchronous; reset if terminal is slave synchronous.
	9		Set if terminal will bid for line first; reset if terminal will not bid for line first.
	10-14		Reserved.
	15	Free Bit	Set if available to user; reset if in use by system.
2	0-15	Secondary Status Word	The bit settings listed below reflect the status of the terminal named in the current terminal name field. The status bits set at the completion of a logical receive status request. Bit 0: Set when the terminal is physically enabled. Bit 1: Set when the terminal is logically enabled. Bit 2: Terminal RECEIVE in progress. Bit 3: Terminal SEND in progress. Bit 4: Reserved. Bit 5: Set when DISABLE is requested. Bit 6: A RECEIVE just completed on the terminal specified in the UTCB prefix. Bit 7: All output data queued for this terminal has been transmitted. Bits 8-11: Reserved. Bits 12-15: These bits denote the type of terminal. The bit settings are 0001 representing a Memorex read/write terminal, 0010 representing TTY 33, 35, 37 or 38 ASR or KSR; 0011 representing a synchronous terminal or processor; 0101 representing a Memorex write only terminal.
4	0-15	Primary Status Word	The bit settings listed below reflect the status of the terminal named in the current terminal name field. The status bits set at the completion of a logical receive status communications operation or a SEND macro operation using the MODE operand's OK, STOP, or NL options. Bits 0-6: Reserved Bit 7: Set when an end-of-message is received at the end of a receive logical operation. This bit is reset when an end-of-unit is received at the end of a receive logical operation. Bit 8: Set when the CNKTIME or RCVTIME timers set by the TRMDEV macro expire. Bit 9: Set when the buffer fills without receiving a termination character. Bit 10: Set when a break or reverse interrupt is received. Bit 11: Set when a transmission error occurs. The data containing the error is moved to the user's work area unedited and untranslated. Bit 12: Set when a modem error is detected.

Displacement		Name	Description
Byte	Bits		
			Bit 13: Set when a line disconnect occurs. Bit 14: Set when an overload causes data to be overlaid before it could be transferred to the processing unit's buffer. This is a hardware lost-data condition. Bit 15: Sets when in synchronous operation and no sync or data is received in three seconds.
6	0-15	Current Data Size	The number of bytes transmitted with a SIGNON or SIGNOFF message.
8	0-63	Current Terminal Name	The logical name of the subject terminal.
16	0-7	No. Terminals-1	The number of terminals named in the UTCB minus one.
	8-15	Current Entry	The number of the current entry in the terminal name list.
18	0-63	First Terminal Name	The name of the first terminal described in the TERMINAL macro TERMNAM operand.
26	0-n	N^{th} Terminal Name	The remainder of the terminal names described in the TERMINAL macro TERMNAM operand.

F. USER WORK AREA PREFIX

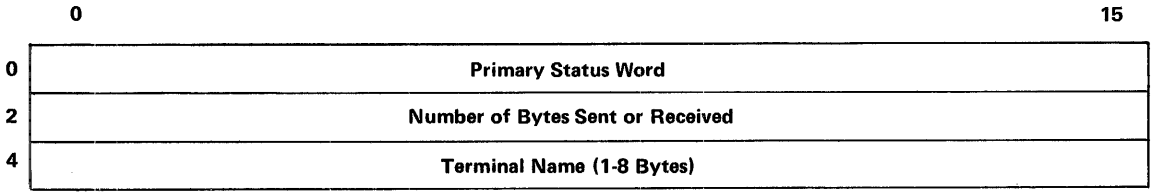
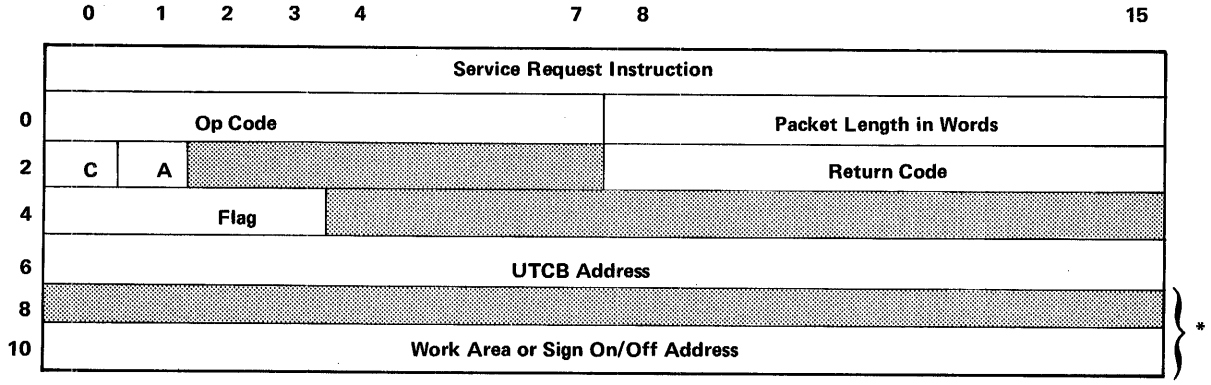


Figure F-1. User Work Area Prefix

The fields in the prefix area are defined in the following table.

Displacement		Name	Description
Byte	Bits		
0	0-15	Primary Status Word	The bit settings listed below reflect the status of the terminal named in the terminal name field. The status word sets at the completion of a logical receive or send operation only if the user codes the RECEIVE macro WKAREA operand or SEND macro WKAREA operand. Bits 0-6: Reserved. Bit 7: Set when an end-of-message is received from the terminal. This bit is reset when an end-of-unit is received from the terminal. Bit 8: Set when the CNKTIME or RCVTIME timers set by the TRMDEV macro expire. Bit 9: Set when the buffer fills without receiving an EOU or EOM termination character. Bit 10: Set when a break or reverse interrupt is received. Bit 11: Set when a transmission error occurs. The data containing the error is moved to the work area unedited and untranslated. Bit 12: Set when a modem error is detected. Bit 13: Set when a line disconnect occurs. Bit 14: Set when an overload causes data to be overlaid before it could be transferred to the processing unit's buffer. This is a hardware lost-data condition. Bit 15: Set when in synchronous operation and no sync or data is received in three seconds.
2	0-15	Bytes Sent or Received	Number of bytes transmitted or received.
4	0-63	Terminal Name	The logical name of the terminal receiving or sending data.

G. SERVICE REQUESTS PARAMETER PACKET FOR THE ENABLE , DISABLE, RECEIVE AND SEND MACROS



***These two words are not included when ENABLE/DISABLE is issued without a SIGNON/SIGNOFF message.**

Figure G-1. Service Request Parameter Packet

The fields of the parameter packet are defined in the following table.

Displacement		Name	Description
Byte	Bit		
0	0-7		The operation codes (in hexadecimal) are as follows: 01 Enable 02 Enable with Sign-on 03 Receive 04 Send 05 Send OK 06 Disable 07 Disable with Sign-off 08 Receive Status 09 Send-STOP 0A Send New Line 0B Stop partition request by CLS
	8-15	Packet Length	Word length of parameter packet.
2	0	Complete Bit	Set upon completion of the SR. If the user codes RETURN=YES, he must check the Complete bit to insure that his request was processed.
	1	Abnormal Bit	Set if the request was abnormally completed.
	2-7		Reserved.
	8-15	Return Code	Set at the completion of a logical communications operation. This field informs the user program as to the success or failure of the communication operation. A field description designating each hexadecimal combination is provided in Table G-1. Table G-2 shows which hexadecimal combination applies to each service request macro.
4	0-7	Flag	Flag bit designations are as follows: Bit 0: Sets when the work area size is in the UTCB prefix field. Bit 1: Reserved.

Displacement		Name	Description
Byte	Bits		
			Bit 2: Set when the RELEASE=YES operand of the DISABLE macro is coded. Resets when the SEND macro MODE operand is omitted or if MODE=EQU is coded. Sets when the SEND macro MODE=EOM operand is coded. Bit 3: Set to indicate a single terminal operation (that is, PREFIX=YES). Bit 4: Indicates transmission to tape. Bit 5: Set to indicate transparent transmission.
	6-15		Reserved.
6	0-15	UTCB Address	The address of the UTCB.
8	0-15		Reserved.
10	0-15	Work Area or Sign On/Off Address	The address of the work area, sign-on, or sign-off transmission.

Table G-1. Service Request Return Code Meanings

Return Code	Description
00	The service request was successfully completed.
D0	The terminal name supplied was not defined by Control Language. This implies a misspelling in the TERMINAL operand or an omission in a Control Language operand.
D1	There are no system buffers available for output. This includes SIGNON or SIGNOFF messages. This code will also be used when the output size exceeds the length of any of the buffers that are available.
D2	There are no buffers available for input. All the system buffers are in use or there are none and a read cannot be put up to the line.
D3	The terminal has already been disabled.
D4	A macro usage error has been detected. The following are examples of this type of error. The terminal to be enabled is already enabled in another partition. The command issued contradicts the mode defined by the TERMINAL macro (for example, issuing RECEIVE to a SEND-only line). First service request was not an ENABLE.
D5	An exception condition was detected. The user should interrogate the Primary Status word for more information. A RECEIVE with STATUS=YES can then be issued to get the Secondary Status word.
D6	A BREAK (asynchronous only) or an RVI (synchronous only) was received.
D8	There is not enough space available in the partition pool for ENABLE to get the memory it needs.
D9	The terminal name asked for with //DEF ID=xxxxxxxx, was not named in a TRMDEV macro which built the TDT descriptions on disc.
DA	Incorrect //DEFINE statements supplied in the Control Language.
DB	Terminal with invalid ID attempted to connect.
DC	Request not honored because job is in termination.
DD	Irrecoverable Block I/O disc read error when reading SYSIN, //DEF tables, or the TDT descriptions from NUCLIB.

The return codes in the following table apply to the macros noted by an X.

Table G-2. Relationship of Macros to Return Codes

Return Code	ENABLE	DISABLE	SEND	RECEIVE
00	X	X	X	X
D0	X	X	X	X
D1	X	X	X	
D2	X			
D3			X	X
D4	X		X	X
D5	X	X	X	X
D6			X	X
D8	X			
D9	X			
DA	X			
DB	X			
DC	X	X	X	X
DD	X			

H. SYNCHRONOUS LINE CONTROL FOR TERMINAL RESPONSE AND MESSAGE FRAMING

LOGICAL COMMUNICATIONS LINE CONTROL FOR RECEIVE MODE

Refer to Figures H-1 and H-2 for outlines of synchronous sending and receiving logic.

TCOM RESPONSES TO A TERMINAL'S ENQ BID

ACK0	TCOM acknowledges the terminal's bid for the line.
WACK	TCOM acknowledges the terminals bid for the line but does not wish to receive input. The terminal should continue to bid for a positive response.
EOT	TCOM does not wish to receive input from the terminal and will attempt to bid for the line.
DLE EOT	TCOM recognized an irrecoverable error condition and is aborting the line connection.
no response	Terminal should continue to bid.

RESPONSES TO TEXT RECEIVED

ACK0 or ACK1	TCOM received the first TRANSUNIT transmitted by the remote terminal without recognizing an error and is ready to receive the next TRANSUNIT. TCOM will send ACK1 or ACK0 alternating the positive responses.
NAK	TCOM wishes the remote terminal to retransmit the last TRANSUNIT.
WACK	TCOM acknowledges the last input but wishes to delay reception of the next input. The terminal should bid (ENQ) before sending the next input. TCOM will respond to the bidding ENQ with a positive response when it is ready to receive text.
EOT	TCOM is aborting message reception due to user program mode error (Send, Receive, Conversational).
DLE EOT	TCOM is aborting message reception as well as the line connection due to an irrecoverable error.
RVI	TCOM wishes to temporarily stop receiving input; however, one additional TRANSUNIT may be sent by the remote terminal.

RESPONSES TO EOT RECEIVED

EOT	TCOM does not wish to transmit, but does not wish to disconnect (Switched only) the line.
DLE EOT	TCOM does not wish to transmit and is disconnecting (Switched only) the line.
ENQ	TCOM is bidding for the line to send output.

TIME-OUTS

TCOM will allow three seconds for the reception of all control sequences and a user specified time for input. If no input is forthcoming in the time allotted, TCOM will send the last output sequence and attempt to reread the input sequence TWICE. If the two retries are unsuccessful, the Time-out bit is set in the Primary Status word. The time allotted for input TRANSUNITs is specified by the user when coding the RCVTIME operand in the TRMDEV macro.

LOGICAL COMMUNICATIONS LINE CONTROL FOR HANDLING TRANSUNITS

	Input	Comments
TRANSUNIT	SYN,SYN	1
	STX	2
	TEXT	3
	ETB	4
	CRC	5
	PAD	6

Comments

1. The SYN sequence is used by the hardware to obtain line synchronization and is deleted from the input character stream.

2. The framing characters are stripped.

3. This sequence is placed into the user's buffer and work areas unedited and untranslated.

4. This sequence will cause the hardware to terminate input after the CRC sequence (Cyclic Redundancy Check) is read. If the character read compares equal to that specified in the EOU operand of the TRMDEV macro, the End of Message bit in the Primary Status word is set to 0. If the character does not compare equal to the EOU operand in the TERMDEV macro, the End of Message bit is set to 1.

5. The CRC sequence is compared to the hardware accumulated CRC. If they compare equal the message is passed to the user. An unequal compare will cause TCOM to

retry the operation a user-specified number of times. If retry is unsuccessful the Transmission Error bit is set in the Primary Status word.

6. The PAD character is required by the modem and is not sent to the Processing Unit buffer area.

LOGICAL COMMUNICATIONS LINE CONTROL FOR SEND MODE

EXPECTED TERMINAL RESPONSES TO TCOM's ENQ BID

ACK0 — This response will cause TCOM to send the first TRANSUNIT to the remote terminal.

WACK — TCOM will continue bidding ENQ until a positive response is received from the terminal.

ENQ — TCOM will rebid ENQ until the retry limit is reached to attempt output to the terminal. If the rebids are unsuccessful, the return codes $D9_{16}$ will be set when the next SEND macro is issued for this terminal.

EOT — $D9_{16}$ will be placed in the return code field when the next SEND macro is issued for this terminal. TCOM will not retry bidding when this response is received.

DLE EOT — The remote terminal has indicated it is disconnecting from the line. TCOM will set the Primary Status End of Message bit and the Line Disconnect bit. This status will be provided when the next SEND macro is issued to this terminal.

Once the status is received by the user, a RECEIVE macro with a STATUS=YES operand should be issued to determine the terminal state.

EXPECTED RESPONSES TO TEXT SENT TO THE TERMINAL

ACK1 — The remote terminal has received the data without detecting an error. TCOM will expect alternating positive responses; that is, ACK0 and ACK1.

WACK — The terminal has acknowledged the data sent by TCOM, but wishes to delay before receiving the next input. TCOM will bid for the positive response to continue output.

NAK — The remote terminal wishes TCOM to retransmit the last transmission.

EOT — The remote terminal is aborting message reception due to user program discipline error.

DLE EOT	The remote terminal is aborting message reception as well as the line connection due to an irrecoverable error.
RVI	The terminal wishes to temporarily stop receiving input and send a priority message. One additional transmission may be sent to the remote terminal.
Unidentifiable response	Send ENQ to have last acknowledgment resent.

TCOM'S INTERPRETATION OF RESPONSES TO EOT SENT

EOT	The terminal does not wish to transmit but does not wish to disconnect (Switched only) the line.
DLE EOT	The terminal does not wish to transmit and is disconnecting (Switched only) the line.
ENQ	The terminal is bidding for the line to send data to TCOM.

TIME-OUTS

TCOM will allow one second for the reception of all control sequences. If no input is forthcoming in the time allotted, TCOM will send the last output sequence and attempt to reread the input sequence until the retry counter is exhausted. If the retries are unsuccessful, the Time-Out bit is set in the Primary Status word.

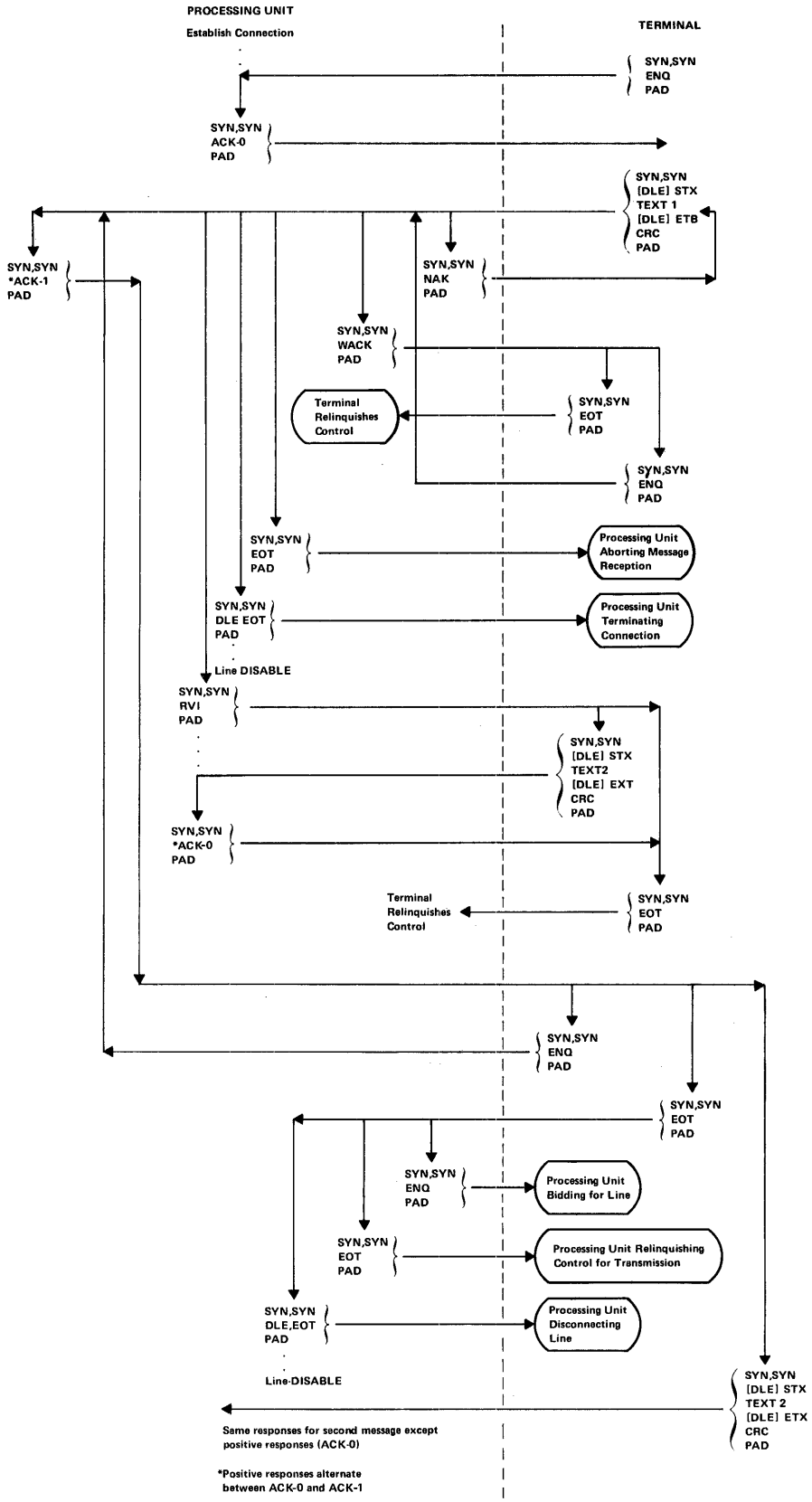


Figure H-1. Synchronous Receiving Logic

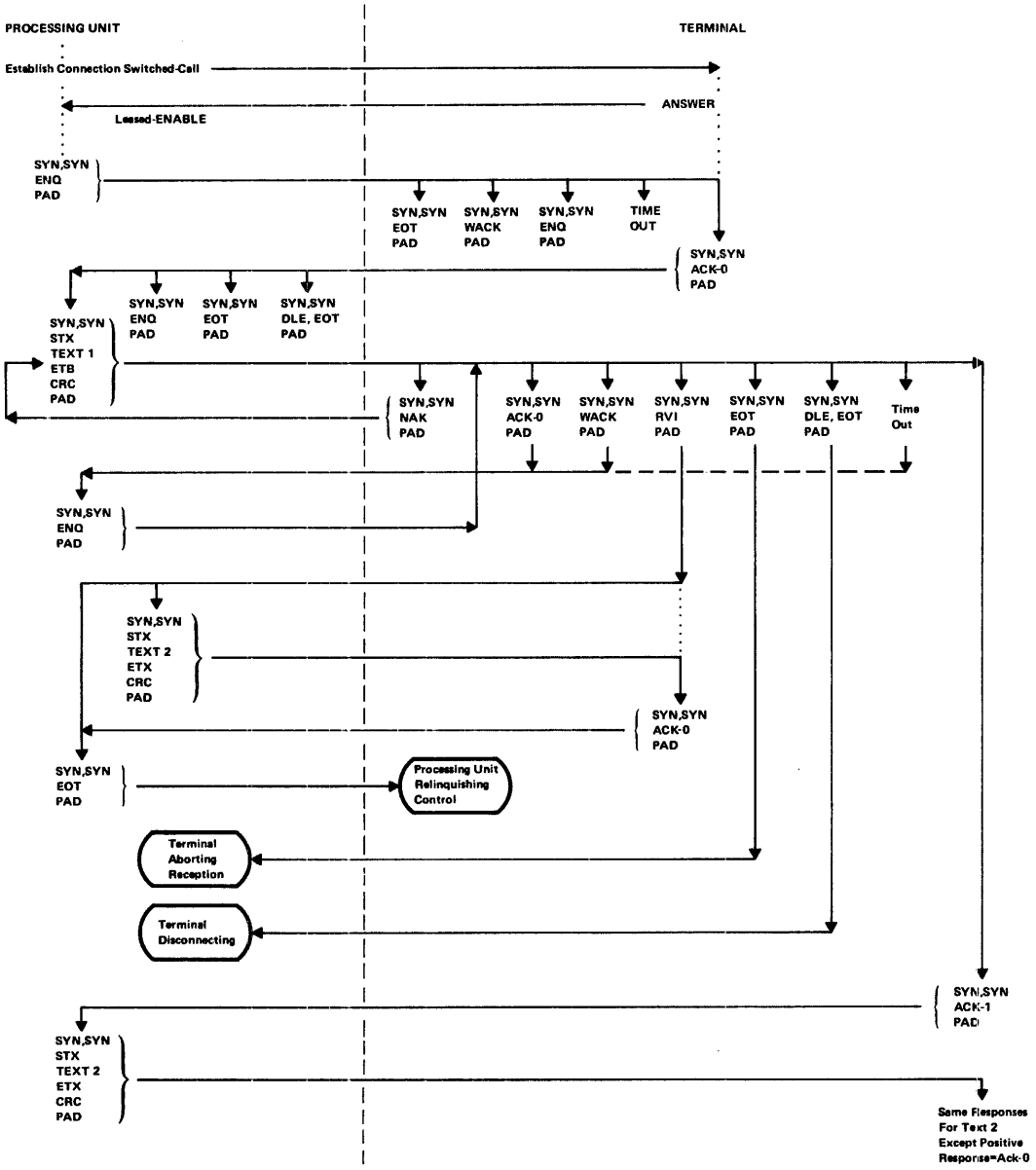


Figure H-2. Synchronous Sending Logic

I. ASYNCHRONOUS LINE CONTROL FOR TERMINAL RESPONSE AND MESSAGE FRAMING

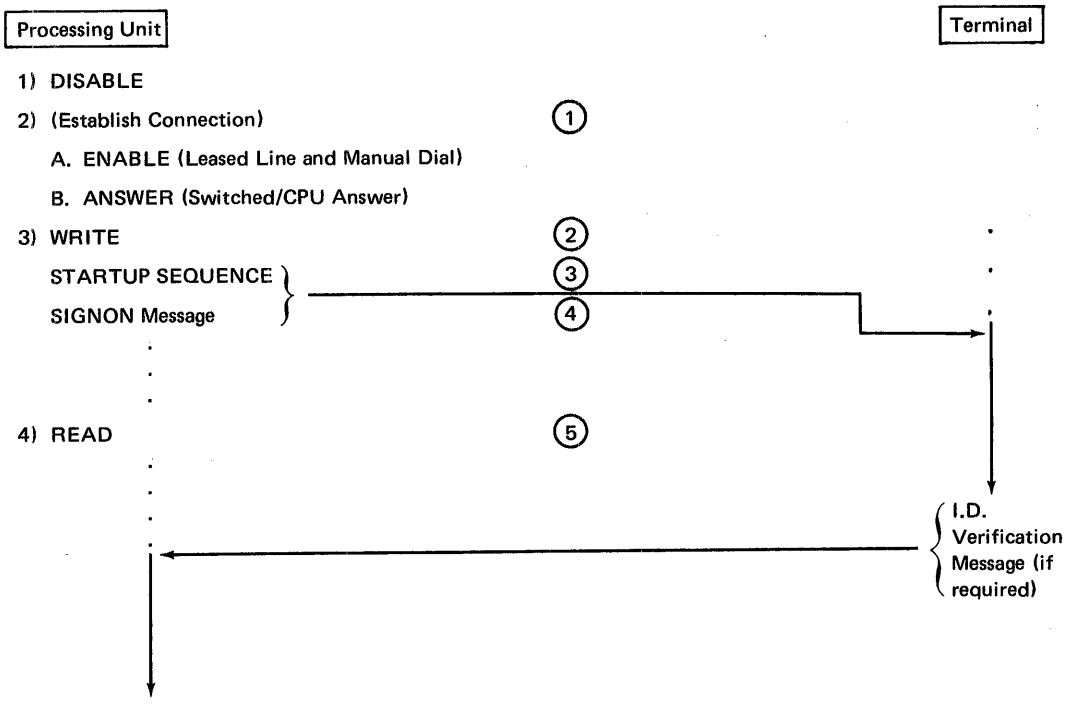


Figure I-1. Asynchronous Non-Switched Line Control to Establish Connection

Comments

(1) The command used for establishing connection will utilize the timer specified in the CNKTIME operand of the TRMDEV macro.

(2) If the TERMINAL macro specifies ATTENDD=NO or the ENABLE macro specifies a SIGNON message, Logical TCOM will issue a write immediately after the line connection is established.

(3) The character sequence defined by the user in the FFRAME operand of the TRMDEV macro will precede the text of the SIGNON message if ATTENDD=NO is specified.

(4) Framing characters will not be contained in the SIGNON message. The operand of the SIGNON keyword is transmitted unedited and untranslated.

(5) If the user defines an ID verification sequence in the TRMDEV macro a read will be issued to the line. This read will follow the SIGNON write or the command which establishes connection if no SIGNON message is requested. The input data will be compared to the sequence defined in the operand field of the ID operand in the TRMDEV macro. The ID will be matched against all terminals assigned to the line group and if no match occurs the line control procedure will be restarted with the

DISABLE command. If no connection can be established after two retries and no other terminal and/or line connections can be established, the ENABLE macro will be completed with a Return Code and a Status bit set in the Primary Status field of the UTCB prefix.

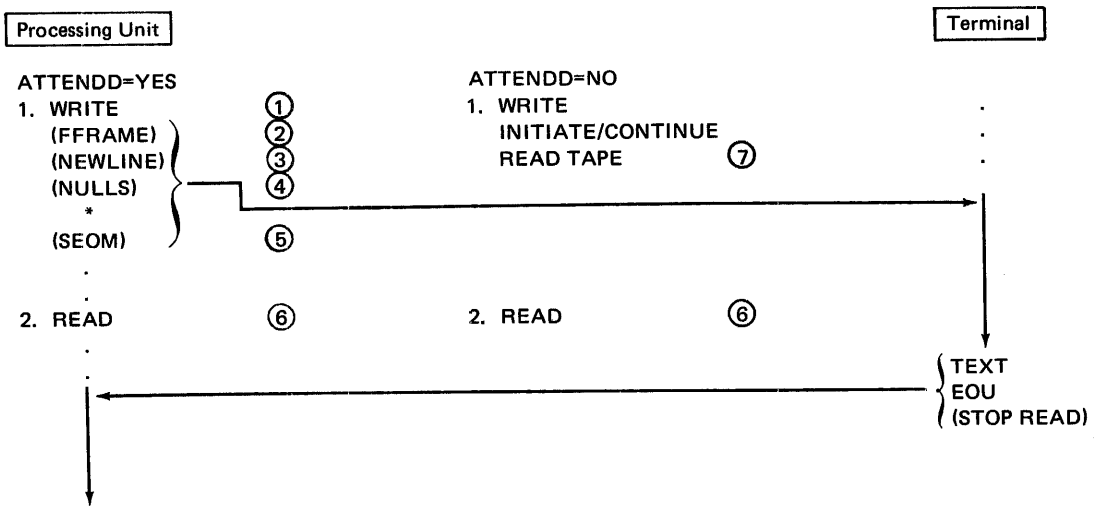


Figure I-2. Asynchronous Non-Switched Line Control for Processing Unit Receiving

COMMENTS

(1) The character sequence defined by the user as the operand of the FFRAME keyword on the TRMDEV macro.

(2) The character sequence defined by the user as the operand of the NEWLINE keyword in the TRMDEV macro.

(3) The number of NULLS inserted between the NEWLINE and the * is defined by the value specified in the NULLS operand of the TRMDEV macro.

(4) An asterisk (*) is transmitted if the TERMINAL macro keyword PROMPT=YES.

(5) The character sequence defined in the SEOM operand of the TRMDEV macro will be sent to the terminal.

(6) All non-EBCDIC messages received without error will be translated and edited. If a hardware termination character is detected in the input text, the read command will be terminated and the termination character will be compared to the character specified by the user in the EOU operand of the TRMDEV macro. If a match is made, the input is treated as a TRANSUNIT. If no match is found and the characters are not a CAN or ENQ, the input is treated as a message.

All read commands must contain a timer specified by the user. This value is defined as the entry in the RCVTIME operand in the TRMDEV macro. When the timer expires, the Processing Unit receipt of input will be terminated.

(7) The appropriate initiate or continue Read Tape character as specified in the operand of the TAPE keyword in the TRMDEV macro.

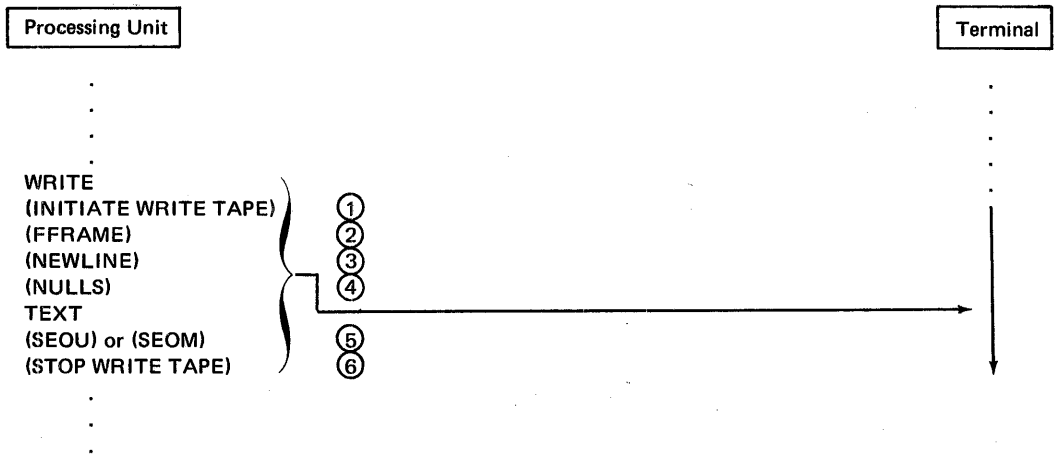


Figure I-3. Asynchronous Non-Switched Line Control for Processing Unit Sending

COMMENTS

(1) The write initiate character sequence is sent to the terminal if TAPE=YES is specified by the user in the SEND macro. The message will then be punched/written on tape and printed on the keyboard.

(2) The character sequence defined as the operand of the TRMDEV macro FFRAME keyword.

(3) The character sequence defined as the operand of the TRMDEV macro NEWLINE keyword.

(4) The number of nulls defined as the operand of the TRMDEV macro NULLS keyword are inserted in the output stream between the NEWLINE sequence and the TEXT.

(5) A two character sequence is sent after the message text. If MODE=UNIT, the SEOU sequence is sent. If MODE=EOM, the SEOM sequence is sent.

(6) If TAPE=YES is specified in the SEND macro, the write stop character is sent to the terminal; all writing/punching is disabled.

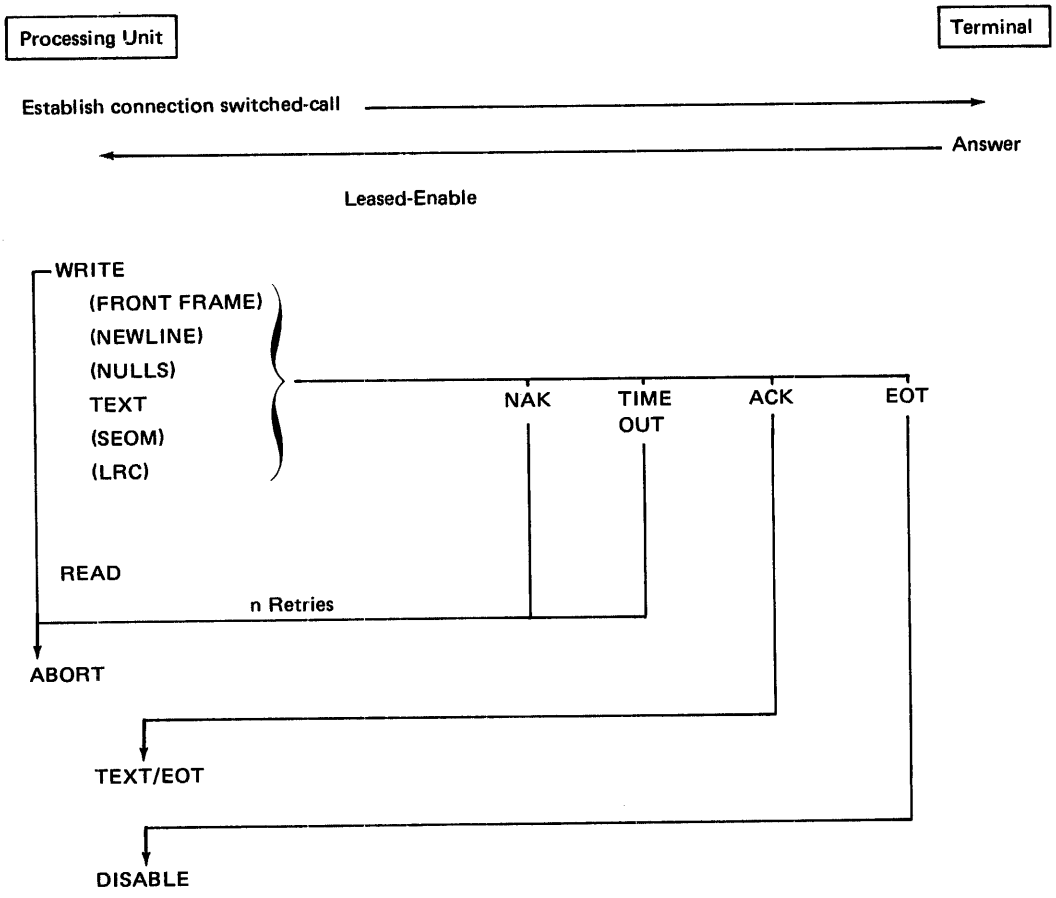


Figure I-4. Asynchronous Send with ASCII Checking

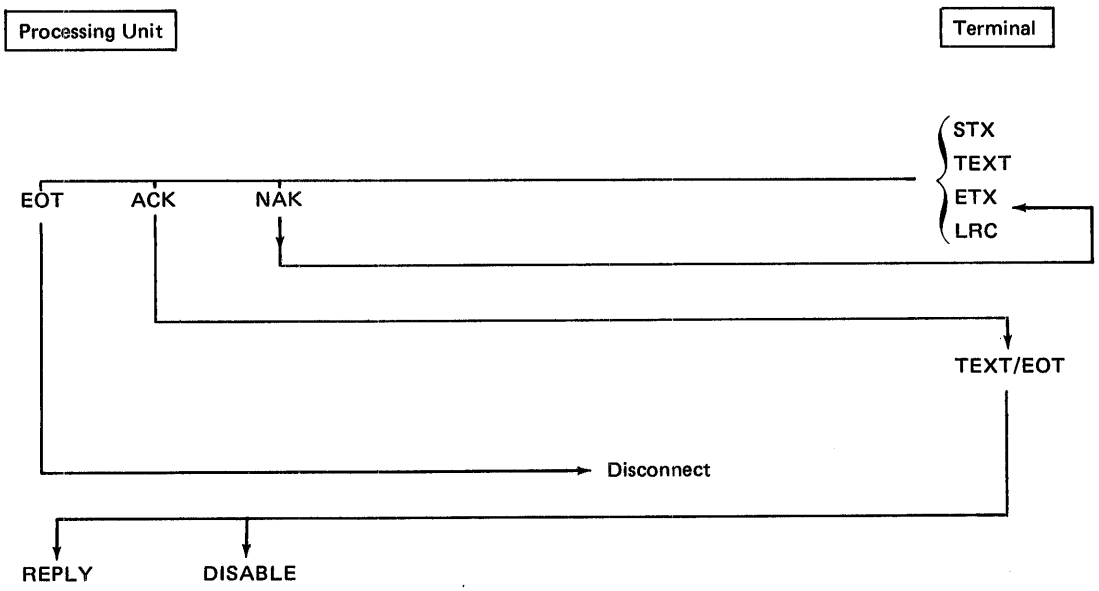


Figure I-5. Asynchronous Receiving with ASCII Checking

J. LOGICAL TCOM ERROR RECOVERY PROCEDURES

RESPONSE TO DETECTABLE ERRORS

This appendix defines Logical TCOM terminal and line error recovery procedures for the terminals supported. If an irrecoverable error occurs during a service request communication operation, it is reported to the user in the parameter packet's Return Code field and the Primary Status field. Otherwise, the Return Code field is filled with hexadecimal zeros. If an irrecoverable error occurs while in error recovery for a recoverable error, the irrecoverable error will be processed. All currently defined errors, with the exception of line disconnect are recoverable.

Depending on the error condition (see Table J-1) the error recovery procedure includes:

- Retry
- Making an entry in the Error Log File
- Operator's console message

The following table lists the error procedure for each error condition.

Table J-1. Actions Taken for Detectable Errors

Type of Error	Command Retried*	Bit Set in Primary Status Word	Return Code in User's Parameter Pocket	Console Message Written	Abnormal and Complete Bits Set in Parameter Packet	Entry on Error Log File
Hardware Lost Data	X	14	D5	X	X	X
Hardware Time-Out (SYN only)	X	15	D5	X	X	X
Line Disconnect		13	D5	X	X	X**
Modem Error	X	12	D5	X	X	X**
Software Lost Data		1	D5	X	X	
Software Timer	X	8	D5	X	X	X
Transmission Error	X	11	D5	X	X	X

In this chart X = Yes, a blank space = No.

*The command can be retried by using the RETRIES operand in the TRMDEV macro.

**For these errors, a DISABLE is issued to the line.

RETRY

Retry is another attempt to execute the command or command program. The procedures for retrying reads and writes on synchronous and asynchronous terminals are:

	Synchronous	Asynchronous
READ	1. WRITE a NAK. 2. Reissue READ command.	1. Write TRANSMISSION ERROR, RETRANSMIT to the terminal. 2. Reissue prompting sequence if any. 3. Reissue READ command.
WRITE	1. Reissue WRITE command. 2. Issue READ for acknowledgment.	No procedure; the only possible error, loss of Clear to Send, is not recoverable.

ERROR LOG FILE

The 40-byte entry in the system error log file has the following format.

Byte(s)	Contents	Description
0	X'06'	Record identification.
1		Unused.
2-4	yymmdd	Date.
5-10	hhmmssss	Time.
11-12	xx	Hardware clock.
13	X'xx'	Code to identify the processor in which execution is taking place. X'08' for Processor 0. X'04' for Processor 1. X'02' for Processor 2. X'01' for Processor 3.
14	X'xx'	Physical device address.
15	X'xx'	Failing command code.

Byte(s)	Contents	Description
16-17	X'xxxx'	Primary status word.
18-19	X'xxxx'	Secondary status word.
20-21	X'xxxx'	Number of retries.
22-29	C'xxxxxxxx'	Logical identifier for the terminal that was interacting when the error occurred.
29-30	X'xxxx'	Address of failing command word.
31-39		Reserved.

CONSOLE MESSAGES

Console messages and formats are described with all other error messages in Appendix K.

INVALID SYNCHRONOUS RESPONSE ERROR

If an unrecognizable response is received, an ENQ is written to the line, and a read is issued for a repeat of the response. If the response is NAK, the ENQ is written again. If the response is the expected ACK, the last operation is posted complete and processing continues. If the response is the wrong ACK, Logical TCOM writes an EOT.

K. TCOM ERROR MESSAGES

Three types of messages can be issued during telecommunications processing. They include console error messages, source error messages, and object-time error messages.

CONSOLE ERROR MESSAGES

Irrecoverable transmission errors that occur during a telecommunications program are posted on the console. Since these errors are irrecoverable, the message simply identifies the error without indicating a response to be made for error correction.

Console messages have the following format:

ER TP eee paa termname ssss

where:

ER — indicates that error recovery was unsuccessful.

TP — indicates that the error occurred on a telecommunications device.

eee — is a 3-digit decimal error code that is variously:

- 001 — meaning that the error is hardware-related. Additional information is supplied in the ssss portion of the message. Refer to Table K-1 for an explanation of the hexadecimal status completion code (ssss).
- 002 — meaning that the connect or receive time requested is invalid.
- 003 — meaning that an abort request was issued for a given line from outside its partition.
- 004 — meaning invalid line connection.
- 005 — meaning message format error.

p — identifies the processor using the device (line). For telecommunications errors p is always zero.

aa — is a 2-digit hexadecimal number specifying the physical address of the device (line).

termname — is the logical terminal name assigned by Control Language Services, e.g., the name specified in the ID= keyword parameter on the //DEFINE statement.

ssss — is the hexadecimal status completion code that is defined in Table K-1.

Table K-1. Hexadecimal Status Completion Codes (SSSS)

Hex Status Completion Code (SSSS)	Meaning	Description	Comments
8000	Abort Requested	The command program was aborted via the user's request.	The command in execution is stopped and the communication request is aborted.
4000	Timer Expired	The timer requested with this command expired.	The command in execution is stopped and the communication request is terminated.
2000	Termination Character Received	The communication driver detected a requested termination character in the input data.	The command and the command program are terminated.
1000	BREAK Received	A break was detected on the addressed asynchronous line or an RVI was detected on a synchronous line.	If the command in execution is a WRITET, execution stops immediately. The execution of any other command is not altered. This only applies to asynchronous lines. The RVI is synchronous only and is received as a response.*
0800	Intermediate BCC detected	A BCC status was received and the input buffer was not terminated.	The command terminated when termination character was received. Input buffer contains ESB's.
0400	Software Lost Data	One or more input characters were received by the communication driver and no read command for this line was pending.	The output data transfer command in execution (for 103A only) or the next data transfer command is executed to completion. The command program is then terminated.*
0200	Transmission Error	No stop bit, VRC error, or a BCC/LRC error occurred on the processing unit receipt of data.	The input command is executed to completion and the command program is terminated.
0100	Reserved		
0080	Modem Error	A modem error was detected on the addressed line.	The command in execution and the command program are terminated upon detection of this condition. The user can issue a RECEIVE with STATUS=YES for more information.
0040	Line Disconnect Detected	A line disconnect was detected on the addressed line.	The command in execution and the command program are terminated upon detection of this condition.
0008	Ring Indicator	The ring indicator was received for the addressed line.	This status bit is set only when the RPTRING command is pending. The command program is terminated upon detection of this condition.
0004	Hardware Lost Data	A processing unit overload was detected and the addressed line adapter was unable to enter the assembled character into the common control (hardware) queue.	The command in execution or the next command issued is executed to completion and the command program is then terminated.*
0002	Hardware Timer Expired	The BSC-3 second hardware timer expired.	The command in execution or the next command to be executed will continue to completion. The command program will then be terminated.*

*These conditions can occur simultaneously with each other. Also they may occur with only one of those conditions not marked by an asterisk.

SOURCE ERROR MESSAGES

Errors detected during the assembly of TCOM macros are listed in the source program where the macro appears and by line number (that refers back to the line in the source listing where the error occurred) with all other messages at the end of the Assembler Language source listing. These errors are called MNOTEs.

When the message is imbedded in the source listing, it has one of the following two formats:

- MNOTE F, *** Message text
- MNOTE W, WARNING, Message text

where:

F means fatal error

W means warning error

When fatal errors are detected, no TCOM macros are generated. When warning errors are detected, the TCOM macro is generated with the appropriate field(s) either set to zeros or set to the default entry, if one is specified.

When the message is listed at the end of the Assembler Language source listing, it has the following format:

LINE NUMBER	ERROR TYPE	ERROR CODE	MESSAGE TEXT
nnnnn	t	aappccc	MNOTE ****

where:

nnnn is a 4-digit decimal number specifying the line in the source listing where the error occurred.

t is either W or F designating the error as warning or fatal.

aappccc is a 7-character error code where:

- aa is always AS specifying the Assembler as the source of the error code.
- pp is a 2-digit number specifying the pass in which the error occurred.
- ccc is a 3-digit decimal number specifying the error within the pass.

MNOTE identifies the message as a macro error message.

**** is an identifier that immediately precedes the message text.

EXAMPLES

The source error messages in this section appear in the source listing where the macro is located. The macro name is for cross-reference purposes only. It is not printed when the message is printed.

The following two examples show a TCOM error message as it would appear in both places: in the source listing where the macro appeared, and in the error message listing according to line number.

Example 1, Imbedded Source Listing Format

```
MNOTE W, WARNING, CHAIN KEYWORD INVALID, NO ASSUMED.
```

Example 2, End of Source Listing Format

```
0073 W AS02074 MNOTE **** CHAIN KEYWORD INVALID, NO ASSUMED
****
```

PHYSICAL TCOM MACRO ERROR MESSAGES

The Macro Name entry is for documentation purposes only. It is not printed as part of the message text.

MACRO NAME	MACRO ID	MESSAGE TEXT
EXCP	MNOTE	W, WARNING, RETURN KEYWORD INVALID, NO ASSUMED. The entry was not YES or NO.
EXCP	MNOTE	W, WARNING, ERRCOMP KEYWORD INVALID, NO ASSUMED. The entry was not YES or NO.
PCB	MNOTE	W, WARNING, DEVTYP KEYWORD INVALID, 6 BYTES RESERVED. The keyword was either omitted or the entry was not MT, UR, DISC, or COMM. Six additional bytes will be generated. They contain sense information.
PCB	MNOTE	W, WARNING, ERROPT KEYWORD INVALID, YES ASSUMED. Entry was not YES or NO.

MACRO NAME	MACRO ID	MESSAGE TEXT
PCB	MNOTE	W, WARNING, CPADR AND FUNCTN NOT ALLOWED, IGNORED. An entry was made for the keywords CPADR and FUNCTN. Zeros will be generated in bytes 6 and 7.
PCB	MNOTE	W, WARNING, FUNCTN INVALID, KEYWORD IGNORED. The entry was not ASKATT or REMOVE. Zeros will be generated in bytes 6 and 7.
PCB	MNOTE	F, ******* FATAL ERROR(S), MACRO NOT GENERATED.

NOTE

The same COMMAND macro is used for TCOM or PIO. All the MNOTEs can be generated by the COMMAND macro, but they do not all pertain to TCOM. This statement applies to the remaining messages in this section.

COMMAND	MNOTE	F, *** ERROR, OPCODE KEYWORD MISSING. The keyword was missing.
COMMAND	MNOTE	F, *** ERROR, OPCODE KEYWORD INVALID. The entry was not one of the reserved words for disc or communications or it was not entered in the form, X'nn' for IOC.
COMMAND	MNOTE	W, WARNING, CHAIN KEYWORD INVALID, NO ASSUMED. The entry was not YES or NO.
COMMAND	MNOTE	W, WARNING, SIZERR KEYWORD INVALID, NO ASSUMED. The entry was not YES or NO.
COMMAND	MNOTE	W, WARNING, SKIP KEYWORD INVALID, NO ASSUMED. The entry was not YES or NO.
COMMAND	MNOTE	W, WARNING, FIELD KEYWORD INVALID. The entry was not RZCNT, RNCNT, or HA.
COMMAND	MNOTE	W, WARNING, CANNOT CHAIN A JUMP COMMAND – IGNORED. The entry CHAIN=YES was entered for a JUMP command. CHAIN=NO assumed.

MACRO NAME	MACRO ID	MESSAGE TEXT
COMMAND	MNOTE	W, WARNING, CWADR KEYWORD MISSING. The JUMP command opcode was entered.
COMMAND	MNOTE	W, WARNING, BUFSIZ MUST BE NONZERO AT EXECUTION TIME. The keyword was not entered for an IOC command.
COMMAND	MNOTE	W, WARNING, TIMER KEYWORD INVALID – IGNORED. The entry was out of range.
COMMAND	MNOTE	W, WARNING, LINECOD KEYWORD MISSING/INVALID – IGNORED. The entry was either missing for a SET command or was not ASCEVEN, ASCODD, ASCNO, or EBCDIC.
COMMAND	MNOTE	W, WARNING, SPEED KEYWORD MISSING/INVALID – IGNORED. The entry was either missing for a SET command or was not 1200, 600, 300, 150, or 110.
COMMAND	MNOTE	W, WARNING, CPUXMIT KEYWORD INVALID – PRI ASSUMED. The entry was not PRI or SEC.
COMMAND	MNOTE	W, WARNING, CANNOT CHAIN THIS COMMAND – IGNORED.
COMMAND	MNOTE	W, WARNING, BUFADR KEYWORD MISSING. The command entered requires a buffer address.
COMMAND	MNOTE	W, WARNING, BUFADR1 KEYWORD MISSING. The keyword was missing.
COMMAND	MNOTE	W, WARNING, BUFSIZ KEYWORD MISSING. The command entered requires a buffer size.
COMMAND	MNOTE	W, WARNING, BUFADR2 KEYWORD MISSING. The keyword was missing.
COMMAND	MNOTE	W, WARNING, ENDLIST KEYWORD MISSING/INVALID. A READS command was entered. This command requires a termination list.
COMMAND	MNOTE	W, WARNING, FIELD KEYWORD INVALID – IGNORED. The entry was not HA, RZCNT, or RNCNT for a DCSRCH command.
COMMAND	MNOTE	W, WARNING, FIELD KEYWORD INVALID, RNCNT ASSUMED. The entry was not RNCNT or RZCNT for DCREAD, DCWRIT, or a DCFWRIT command.

MACRO NAME	MACRO ID	MESSAGE TEXT
COMMAND	MNOTE	W, WARNING, HOME ADDRESS GENERATED, OTHERS IGNORED. COUNT, KEY, or DATA keywords entered with home address.
COMMAND	MNOTE	W, WARNING, CNTSIZ, KEYSIZ, or DATSIZ NOT SPECIFIED. A read without transfer (SKIP=YES) was specified. At least one keyword must be entered.
COMMAND	MNOTE	W, WARNING, HABUF SPECIFIED, ALL OTHERS IGNORED. COUNT, KEY, or DATA keywords were entered with home address.
COMMAND	MNOTE	W, WARNING, HASIZ KEYWORD MISSING. The HASIZ keyword was missing.
COMMAND	MNOTE	W, WARNING, BUFSIZ SPECIFIED, CNTBUF/KEYBUF IGNORED. A multiple block read applies to a data field only.
COMMAND	MNOTE	W, WARNING, CNTBUF, KEYBUF, OR DATBUF NOT SPECIFIED.
COMMAND	MNOTE	W, WARNING, CNTSIZ KEYWORD MISSING. The keyword was missing.
COMMAND	MNOTE	W, WARNING, KEYSIZ KEYWORD MISSING. The keyword was missing.
COMMAND	MNOTE	W, WARNING, DATSIZ KEYWORD MISSING. The keyword was missing.
COMMAND	MNOTE	W, WARNING, DATBUF KEYWORD MISSING. The keyword was missing.
COMMAND	MNOTE	W, WARNING, GAP KEYWORD INVALID. The entry was out of range.
COMMAND	MNOTE	F, *** ERROR, NO COMMAND WORD GENERATION. A fatal error occurred.

LOGICAL TCOM MACRO ERROR MESSAGES

MACRO NAME	MACRO ID	MESSAGE TEXT
TERMINAL	MNOTE	F, *** ERROR, NAME FIELD MISSING. The name field was missing.

MACRO NAME	MACRO ID	MESSAGE TEXT
TERMINAL	MNOTE	F, *** ERROR, TERMNAM KEYWORD MISSING. The keyword was missing.
TERMINAL	MNOTE	F, *** ERROR, CONVERS AND RECEIVE/SEND INVALID. SEND or RECEIVE cannot be entered with CONVERS.
TERMINAL	MNOTE	F, *** ERROR, CONVERS KEYWORD INVALID.
TERMINAL	MNOTE	F, *** ERROR, RECEIVE KEYWORD INVALID.
TERMINAL	MNOTE	F, *** ERROR, SEND KEYWORD INVALID. The entry was not MESSAGE or TRANSUNIT.
TERMINAL	MNOTE	F, *** ERROR, CONVERS OR RECEIVE/SEND MISSING. At least one keyword must be entered.
TERMINAL	MNOTE	F, *** ERROR, PROMPT KEYWORD INVALID. Entry was not YES or NO.
TERMINAL	MNOTE	F, *** ERROR, CONTROL KEYWORD INVALID. Entry was not MI, MN, SI, or SN.
TERMINAL	MNOTE	F, *** FATAL ERROR(S), MACRO NOT GENERATED. Printed if any of the above errors are detected.
ENABLE	MNOTE	F, ***** ERROR, LIST KEYWORD INVALID. The entry was not YES, NO, or null.
ENABLE	MNOTE	F, *** ERROR, TERMINAL KEYWORD MISSING. The keyword was missing.
ENABLE	MNOTE	F, ***** ERROR, TERMINAL KEYWORD INVALID FOR LIST=YES. The entry was not the name of the TERMNAL macro.
ENABLE	MNOTE	F, *** ERROR, SIGNON KEYWORD INVALID. The keyword did not have 2 entries enclosed in parentheses and separated by a comma.
ENABLE	MNOTE	F, ***** ERROR, SIGNON KEYWORD INVALID FOR LIST=YES. The entry was not data-name, integer, or YES.
ENABLE	MNOTE	F, *** ERROR, PREFIX KEYWORD INVALID. The entry was not YES or NO.
ENABLE	MNOTE	W, WARNING, RETURN KEYWORD INVALID, NO ASSUMED. The entry was not YES or NO.
ENABLE	MNOTE	F, *** FATAL ERROR(S), MACRO NOT GENERATED. One or more of the above fatal errors were detected.

MACRO NAME	MACRO ID	MESSAGE TEXT
DISABLE	MNOTE	F, ***** ERROR, LIST KEYWORD INVALID. The entry was not YES, NO, or null.
DISABLE	MNOTE	F, *** ERROR, TERMINAL KEYWORD MISSING. The keyword was missing.
DISABLE	MNOTE	F, ***** ERROR, TERMINAL KEYWORD INVALID FOR LIST=YES. The entry was not the name of the TERMNAL macro.
DISABLE	MNOTE	F, *** ERROR, SIGNOFF KEYWORD INVALID. The keyword did not have 2 entries enclosed in parentheses and separated by a comma.
DISABLE	MNOTE	F, ***** ERROR, SIGNOFF KEYWORD INVALID FOR LIST=YES. The entry was not data-name, integer, or YES.
DISABLE	MNOTE	F, *** ERROR, PREFIX KEYWORD INVALID. The entry was not YES or NO.
DISABLE	MNOTE	F, ***** ERROR, TAPE KEYWORD INVALID. The entry was not YES or NO.
DISABLE	MNOTE	F, *** ERROR, RELEASE KEYWORD INVALID. The entry was not YES or NO.
DISABLE	MNOTE	W, WARNING, RETURN KEYWORD INVALID, NO ASSUMED. The entry was not YES or NO.
DISABLE	MNOTE	F, *** FATAL ERROR(S), MACRO NOT GENERATED. One or more of the above fatal errors were detected.
SEND	MNOTE	F, ***** ERROR, LIST KEYWORD INVALID. The entry was not YES, NO, or null.
SEND	MNOTE	F, *** ERROR, TERMINAL KEYWORD MISSING. The keyword was missing.
SEND	MNOTE	F, ***** ERROR, TERMINAL KEYWORD INVALID FOR LIST=YES. The entry was not the name of the terminal.
SEND	MNOTE	F, *** ERROR, WKAREA NOT SPECIFIED FOR THIS MODE. The keyword was omitted when MODE=UNIT or EOM.
SEND	MNOTE	W, WARNING, WKAREA KEYWORD IGNORED FOR THIS MODE. The keyword was entered when MODE=OK, NL or STOP.

MACRO NAME	MACRO ID	MESSAGE TEXT
SEND	MNOTE	F, ***** ERROR, TAPE KEYWORD INVALID. The entry was not YES or NO.
SEND	MNOTE	F, *** ERROR, MODE KEYWORD INVALID. The entry was not UNIT, EOM, OK, STOP or NL.
SEND	MNOTE	F, *** ERROR, WKAREA KEYWORD INVALID. The keyword was entered in the wrong format.
SEND	MNOTE	W, WARNING, RETURN KEYWORD INVALID, NO ASSUMED. The entry was not YES or NO.
SEND	MNOTE	F, ***** ERROR, TRANSPT KEYWORD INVALID. The entry was not YES or NO.
SEND	MNOTE	F, *** FATAL ERROR(S), MACRO NOT GENERATED. One or more of the above fatal errors occurred.
RECEIVE	MNOTE	F, ***** ERROR, LIST KEYWORD INVALID. The entry was not YES, NO, or null.
RECEIVE	MNOTE	F, *** ERROR, TERMINAL KEYWORD MISSING. The keyword was missing.
RECEIVE	MNOTE	F, *** ERROR, STATUS KEYWORD INVALID. The entry was not YES or NO.
RECEIVE	MNOTE	W, WARNING, WKAREA IGNORED IF STATUS=YES. The WKAREA keyword was entered. STATUS has priority.
RECEIVE	MNOTE	F, *** ERROR, WKAREA KEYWORD MISSING. No entry and STATUS=NO.
RECEIVE	MNOTE	W, WARNING, RETURN KEYWORD INVALID, NO ASSUMED. The entry was not YES or NO.
RECEIVE	MNOTE	F, *** FATAL ERROR(s), MACRO NOT GENERATED. One or more of the above fatal errors occurred.
TRMDEV	MNOTE	F, ******* ERROR, TERMINAL NAME MISSING. TRMDEV terminal name is missing.
TRMDEV	MNOTE	F, ******* ERROR, DEVICE KEYWORD INVALID/MISSING. TRMDEV DEVICE entry not M1240, MRX, TTY, TTY KSR, SYN, or MRXW/O.
TRMDEV	MNOTE	F, ******* ERROR, TRANSIZ KEYWORD MISSING. TRMDEV TRANSIZ entry missing.

MACRO NAME	MACRO ID	MESSAGE TEXT
TRMDEV	MNOTE	F, ******* ERROR, LINTYP INVALID/MISSING. TRMDEV LINTYP must be one of those listed in Table 2-1.
TRMDEV	MNOTE	F, ******* ERROR, CNKTIME INVALID. TRMDEV timer integer is invalid.
TRMDEV	MNOTE	F, ******* ERROR, RCVTIME KEYWORD INVALID. TRMDEV timing for logical read commands is invalid.
TRMDEV	MNOTE	F, ******* ERROR, RETRIES KEYWORD INVALID. TRMDEV retry specification not in 0-15 range.
TRMDEV	MNOTE	F, ******* ERROR ID KEYWORD INVALID. Must be 1-22 EBCDIC or hexadecimal characters.
TRMDEV	MNOTE	F, ******* ERROR, DIAL KEYWORD INVALID. Entry must be 1-11 numeric digits.
TRMDEV	MNOTE	F, ******* ERROR, TRANSLAT KEYWORD INVALID. The entry must be YES or NO.
TRMDEV	MNOTE	F, ******* ERROR, LINECOD KEYWORD INVALID. The entry is not ASCNO, ASCEVEN, ASCODD, or EBCDIC.
TRMDEV	MNOTE	F, ******* ERROR, SPEED KEYWORD INVALID. The entry is not 110, 150, 300, or 600.
TRMDEV	MNOTE	F, ******* ERROR, EOU KEYWORD INVALID. The entry is not a valid termination character.
TRMDEV	MNOTE	F, ******* ERROR, FFRAME KEYWORD INVALID. The entry does not specify two front framing characters.
TRMDEV	MNOTE	F, ******* ERROR, SEOU KEYWORD INVALID. The entry does not specify two trailing TRANSUNIT characters.
TRMDEV	MNOTE	F, ******* ERROR, SEOM KEYWORD INVALID. The entry does not specify two trailing MESSAGE or final TRANSUNIT characters.
TRMDEV	MNOTE	F, ******* ERROR, NEWLINE KEYWORD INVALID. This entry must specify two characters to be transmitted to the terminal to space it up one line.
TRMDEV	MNOTE	F, ******* ERROR, ENDLIST KEYWORD INVALID. Invalid termination character was specified.
TRMDEV	MNOTE	F, ******* ERROR, TAPE KEYWORD INVALID. Invalid read initiate, read continue, write initiate or stop write characters were specified.

MACRO NAME	MACRO ID	MESSAGE TEXT
TRMDEV	MNOTE	F, ******* ERROR, LRC KEYWORD INVALID. The entry was not YES or NO.
TRMDEV	MNOTE	F, ******* ERROR, FATAL ERROR(S), THIS TDT NOT GENERATED ***** No terminal description table was built.

OBJECT-TIME ERROR MESSAGES

There is only one object time error message for telecommunications programming, defined as follows.

If the processor detects a transmission error when an asynchronous terminal is in the receive or conversational mode, logical TCOM posts on the terminal "TRANSMISSION ERROR, RETRANSMIT", reissues the prompting sequence if one is specified, and reissues the applicable command. Data terminals in the send mode (terminals that will only receive data) follow the same retry procedures when a transmission error is detected; however, the terminal message, "TRANSMISSION ERROR, RETRANSMIT", is not posted. Data terminals which are unattended receive TRANSMISSION ERROR.

NOTE

All detectable, irrecoverable errors except for the invalid binary synchronous response error, are posted on the operator console (see *Console Error Messages* in this section) are reported to the user in the return code field and the primary status field of the parameter packet at the completion of the service request communications operations. For complete information on the return code field and primary status field, and error recovery procedures, refer to Appendix J in this manual.

INDEX

COMMENTS FORM

MRX/OS Telecommunications Reference Manual (2200.007-01)

Please send us your comments, to help us produce better publications. Use the space below to qualify your responses to the following questions, if you wish, or to comment on other aspects of the publication. Please use specific page and paragraph/line references where appropriate. All comments become the property of the Memorex Corporation.

	Yes	No
• Is the material:		
Easy to understand?	☐	☐
Conveniently organized?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Accurate?	☐	☐
Suitable for its intended audience?	☐	☐
Adequately indexed?	☐	☐

• For what purpose did you use this publication? (reference, general interest, etc.)

_____

• Please state your department's function: _____

_____

• Please check specific criticism(s), give page number(s), and explain below:

☐ Clarification on page(s) _____

☐ Addition on page(s) _____

☐ Deletion on page(s) _____

☐ Error on page(s) _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

MEMOREX

First Class

Permit No. 14831
Minneapolis,
Minnesota 55427

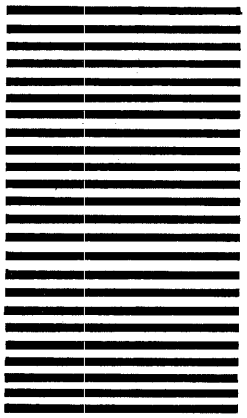
Business Reply Mail

No Postage Necessary if Mailed in the United States

Postage Will Be Paid By

Memorex Corporation

Midwest Operations – Publications
8941 Tenth Avenue North
Minneapolis, Minnesota 55427



. . . . . .

. . . . . .

Thank you for your information. . . . . .

Our goal is to provide better, more useful manuals, and your comments will help us to do so.

. . . . . .Memorex Publications