File No. S360-30
Order No. GC30-2004-6

OS

IBM Systems Reference Library

IBM System/360 Operating System
Basic Telecommunications Access Method

Program Number 360S-CQ-513

This publication describes the Basic Telecommunications Access Method (BTAM) available with Release 20.6/20.7 and Release 21 of the System/360 Operating System combined with the Independent Component Release containing BTAM support for the IBM 3270 Display System. BTAM provides facilities that enable an assembler-language programmer to write a teleprocessing control program that effects communications at the Read/Write level between a System/360 and a variety of computers and terminals connected to the System/360 over common-carrier or private-wire communications networks. BTAM provides similar facilities for the local IBM BTAM 3270 Display System. BTAM employs both start-stop and binary synchronous communications (BSC) techniques, depending on the type of remote station.

Typical BTAM applications include data acquisition, message switching, and inquiry processing.

The publication explains some concepts of teleprocessing and BTAM, describes line control and message transmission techniques, and describes each of the BTAM macro instructions and facilities needed to construct a control program. The READ and WRITE macro instructions applicable for each type of remote station and line configuration are given, along with the channel programs generated for each type.

Prerequisite to use of this publication is a knowledge of System/360 assembler language and data management facilities.

Appendix J of this publication lists the types of terminals that are supported by the Basic Telecommunications Access Method component of the System/360 Operating System.

Terminals which are equivalent to those explicitly supported may also function satisfactorily. The customer is responsible for establishing equivalency. IBM assumes no responsibility for the impact that any changes to the IBM-supplied products or programs may have on such terminals.

Seventh Edition (September 1972)

This edition, GC30-2004-6, is a revision of GC30-2004-5 and associated Technical Newsletters GN30-2551, GN30-2563, GN30-2568, GN30-2569, GN30-2570, and GN30-2571. This edition applies to OS Release 20.6/20.7 and Release 21 combined with the Independent Component Release containing BTAM support for the IBM 3270 Display System.

Significant changes or additions to the specifications contained in this publication are continually being made. When using this publication in connection with the use of IBM equipment, check the latest SRL Newsletter for revisions or contact the local IBM branch office.

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

A form is provided at the back of this publication for reader's comments. If the form has been removed, comments may be addressed to IBM Corporation, Dept. 636, Neighborhood Road, Kingston, New York 12401.

This publication describes the BTAM facilities and macro instructions needed to write an application program that defines, activates, and controls a System/360-based teleprocessing system or local 3270 display system or both. Effective use of this publication does not presuppose a knowledge of teleprocessing techniques, but it does require a knowledge of the System/360 assembly language and of data management techniques. The publication does not contain detailed information on the terminal equipment and computers that may be used as remote or local stations in a BTAM-controlled system. For this information, see the listing of publications below.

This publication is organized as follows:

- Teleprocessing and BTAM Concepts explains concepts of teleprocessing in general, and of BTAM in particular, for the reader who is not already familiar with these concepts. This chapter also defines many terms used throughout the publication.

- Defining the TP System tells how to define to the Operating System the characteristics of the communications lines and equipment comprising the teleprocessing system. It includes information on the system generation procedure. The DCB macro instruction, among others, is given here.

- Buffer Management tells how to construct buffer pools and how to obtain and release buffers as needed to accommodate message data.

- Code Translation describes the facilities BTAM provides for accomplishing the necessary translation between the transmission code used on communications lines and the internal code of System/360 (EBCDIC). The ASMTRTAB and TRNSLATE macro instructions are explained here.

- Activating and Deactivating the TP System tells what procedures to follow in initializing the user program prior to message transmission, and in deactivating the system upon conclusion of message transmission. The OPEN, LOPEN, and CLOSE macro instructions appear here.

- Line Control and Message Transmission briefly explains the techniques for controlling communications lines of various kinds (switched, nonswitched, contention, polling), and presents the READ and WRITE macro instructions, used in message transmission operations. Also given are the WAIT, TWAIT, and RESETPL macro instructions.

- Start-Stop Read and Write Operations lists the types of Read and Write operations applicable to each type of remote terminal, and gives the channel program for each.

- BSC Read and Write Operations lists the types of Read and Write operations applicable to each kind of line configuration, gives the channel program for each, and indicates the types of remote stations for which each type can be used.

- Local 3270 Display System Read and Write Operations lists the types of Read and Write operations applicable to the local 3270 display system and gives the channel program for each type.

- Error Recovery Procedures and Error Recording explains the BTAM-provided facilities for diagnosing and attempting to recover from a variety of error conditions, and for indicating and recording the occurrence of these errors.

- On-Line Testing describes the facilities available for diagnosing line and equipment troubles.

- Sixteen appendixes appear at the back of the publication; these show control block, information table, macro instruction, and error message formats, and code charts.

Before using this manual, the reader should be familiar with the following publications:

<u>OS Assembler Language</u>, GC28-6514

<u>IBM System/360 Operating System: Supervisor Services and Macro Instructions</u>, GC28-6646

<u>OS Data Management Services Guide</u>, GC26-3746

<u>OS Data Management Macro Instructions</u>, GC26-3794

The BTAM user will also need the level of knowledge of information contained in the following publications that apply to the transmission control units and terminals in his equipment configuration:

- Transmission Control Units:

  IBM 2701 Data Adapter Unit, Component Description, GA22-6864

  IBM System/360 Component Description: IBM 2702 Transmission Control, GA22-6846

  IBM System/360 Component Description: IBM 2703 Transmission Control, GA27-2703

- Start-Stop Terminals:

  IBM 1030 Data Collection System, GA24-3018
  IBM 1050 Reference Digest, GA24-3020
  IBM 1050 System Summary, GA24-3471
  IBM 1050 Data Communications System, Principles of Operation, GA24-3474
  IBM 1050 Operator's Guide, GA24-3125

  IBM 1060 Data Communications System, GA24-3034

  IBM System/360 Component Description: IBM 2260 Display Station; IBM 2848 Display Control, GA27-2700

  IBM System/360 Component Description: IBM 2265 Display Station; IBM 2846 Display Control, GA27-2731

  IBM 2740 Communications Terminal, GA24-3403

  IBM 2740/2741 Communications Terminal Operator's Guide, GA27-3001

  IBM 2760 Optical Image Unit Component Description, GA27-3011

- BSC Stations:

  IBM SRL General Information -- Binary Synchronous Communications, GA27-3004

  IBM System/3 RPG-II Telecommunications Programming Reference Manual, SC21-7507

  IBM 1130 Functional Characteristics, GA26-5881

  IBM 1130 Synchronous Communications Adapter Subroutines, GC26-3706

  IBM 2770 System Components, GA27-3013

  IBM 2780 Data Transmission Terminal, Component Description, GA27-3005

  IBM 2790 Data Communication System, Component Description, GA27-3015

  IBM 2972 Models 8 and 11 General Banking Terminal System, Component Description, GL27-3020

  IBM 50 Magnetic Data Inscriber, Component Description, GA27-2725

  IBM 3270 Information Display System, Component Description, GA27-2749

  IBM 3735 Programmable Buffered Terminal Concept and Application, GA27-3043

  IBM 3735 Programmer's Guide, GC30-3001

- Local 3270 Display System:

  IBM 3270 Information Display System, Component Description, GA27-2749

To assemble, linkage edit and execute a BTAM program requires knowledge of the information in:

OS Linkage Editor and Loader, GC28-6538

OS System Generation, GC28-6554

This chapter explains some fundamental aspects of computer-based data communications systems (often called teleprocessing systems) of the kind accommodated by the IBM System/360 Basic Telecommunications Access Method (BTAM), and explains some basic terminology used throughout the publication. As this discussion is intended to explain teleprocessing (TP) systems for the BTAM user, it does not attempt to encompass all kinds of TP systems. Thus, while some of the terms defined apply to all or most communications systems, other terms are limited to communications programming usage, or specifically to BTAM. Moreover, concepts and terminology are presented from the programmer's viewpoint, rather than from the engineer's.

Viewed in its most elementary aspect, a teleprocessing system consists of (1) a central computer and associated transmission control equipment, (2) remote stations, and (3) the electrical circuits (called communication lines or data links) that connect the remote stations to the central computer (See Figure 1). For the purpose of this discussion, the central computer equipment comprises the central processing unit (CPU) and the equipment by which the CPU is connected to the communications lines. The generic name of this equipment is transmission control unit (TCU).

The equipment constituting a remote station can be either a terminal or another computer. A terminal consists of a control unit and one or more input and output devices, each of which is called a component of that terminal. Each input device and each output device is considered a separate component.

Remote stations in a BTAM-controlled teleprocessing system are usually separated from the central computer by a distance sufficient to require common carrier facilities and transmission techniques to accomplish communication between central computer and remote stations. (Communications common carriers are companies that furnish communications services to the public.) However, it is the method of connection to the central computer, rather than the distance from the computer, that determines whether a station is classed as remote. A station is considered remote if it is connected to the central computer through a transmission control unit (TCU). (A station connected directly to a computer data channel is termed a local station.)

Except for the local 3270 display system, the System/360 Operating System version of BTAM supports only remote stations, which must be connected to the central computer by means of an IBM 2701 Data Adapter Unit or an IBM 2702 or 2703 Transmission Control. Local 3270 display systems are connected directly to a selector, multiplexer, or block multiplexer channel of the central computer.

An operator's console is an input/output device whose function is to control the operations of the computer.

The console and its terminal control unit make up a terminal that can communicate with the operating system and with problem programs but cannot communicate with other terminals. If the operating system includes the Multiple Console Support (MCS) option, BTAM can communicate with those operator's consoles that are connected to the central computer through a 2701, 2702, or 2703 transmission control unit.

CATEGORIES OF COMMUNICATIONS LINES

Communications lines can be categorized by several sets of attributes, some of which are discussed below. Some attributes have significance for the user's BTAM program, others need only be specified at system generation time, similar to the way in which the programmer specifies the attributes of local I/O devices.

LINE AND STATION CONFIGURATIONS

A communications line can be classified according to whether it connects two or more than two stations, and whether or not the electrical connection between the central computer and the station is continuously established. Figure 2 illustrates a teleprocessing system comprising several types of line and station configurations, the elements of which are explained below.

A nonswitched line is one that continuously links the stations associated with it, regardless of the amount of time it is in use for message traffic. This kind of line is usually furnished by a common carrier on a contractual basis, between specified locations for a continuous period, or regularly recurring periods, for the exclusive use of one customer.

A nonswitched line is called point-to-point if it connects the computer to a single remote station; or multipoint, if several remote stations are connected to the line.

A switched line is one in which an electrical connection between the central computer and a remote station is established by dialing, similar to the manner in which ordinary telephone calls are made. As in the public telephone network, the actual communication path for a given transmission is not fixed, but is automatically selected from a variety of possible paths by common carrier switching equipment.

Each remote station on a switched line is continuously connected to the common carrier switching center (exchange) by an access line in the same way as a telephone. A telephone number is associated with the access line. Similarly, each transmission control unit at the central computer is connected to the exchange by access lines. Usually, a TCU has several access lines, each with its own telephone number; multiple access lines permit simultaneous communication with several remote stations. Each connection of an access line at the TCU is called a switched line termination, or line appearance.

Common carriers usually charge for switched lines on a time-used rather than a contractual basis.

A switched line is always considered point-to-point, as communication occurs with only one remote station on a line during any call. Switched line connections are established by manual dialing, or by automatic dialing under program control. The dialing operation may be performed at the central computer or at the remote station, and the called station answers manually or automatically. Not all of these options are available for all types of line configurations and remote stations. Within the limitations imposed by equipment or programming, the user chooses between them on the basis of the requirements of his application. For example, if the application involves collection of batched data from a number of remote stations after normal working hours, it would be appropriate to have each station equipped with an automatic answering facility and the central computer equipped with the calling facility. This would allow the program automatically to call the unattended stations and receive the batched data.

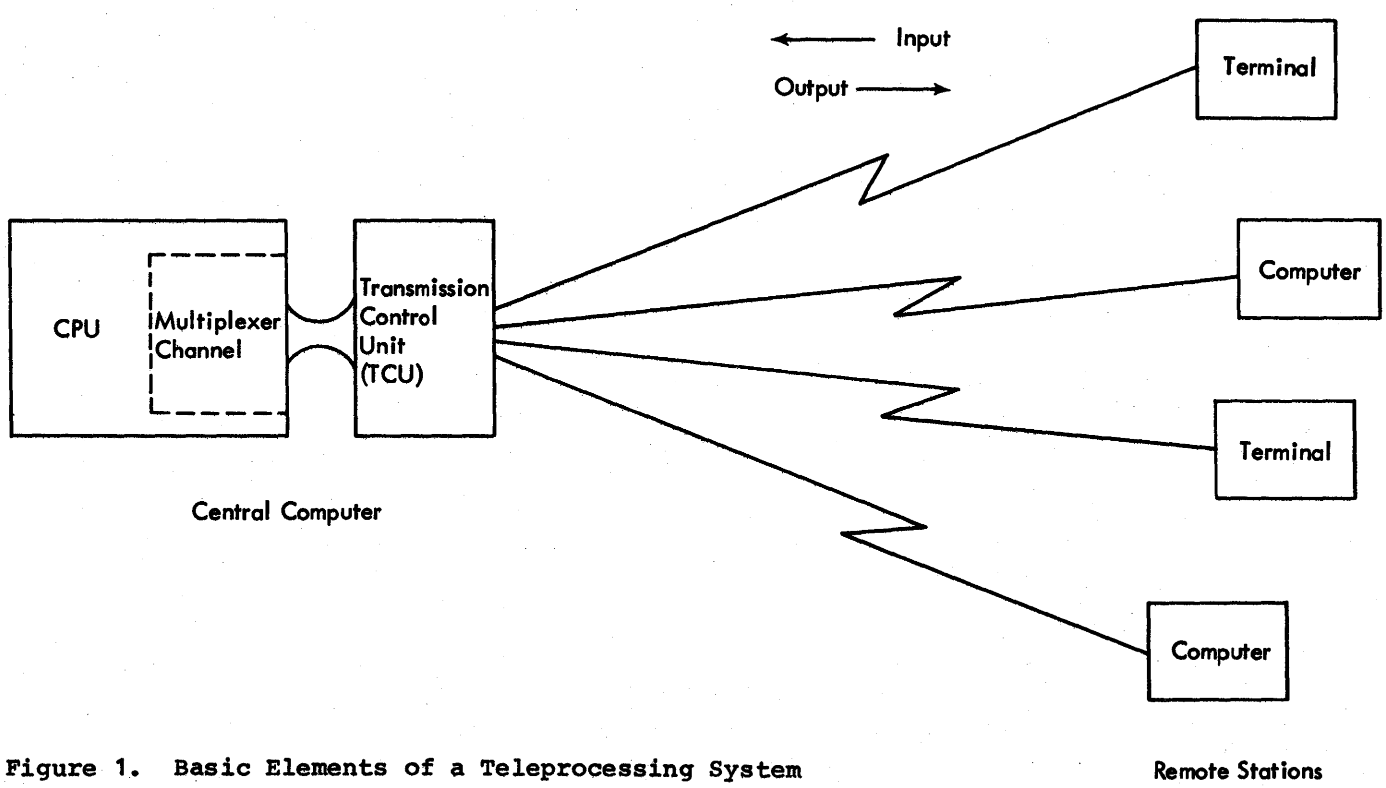


Figure 1. Basic Elements of a Teleprocessing System

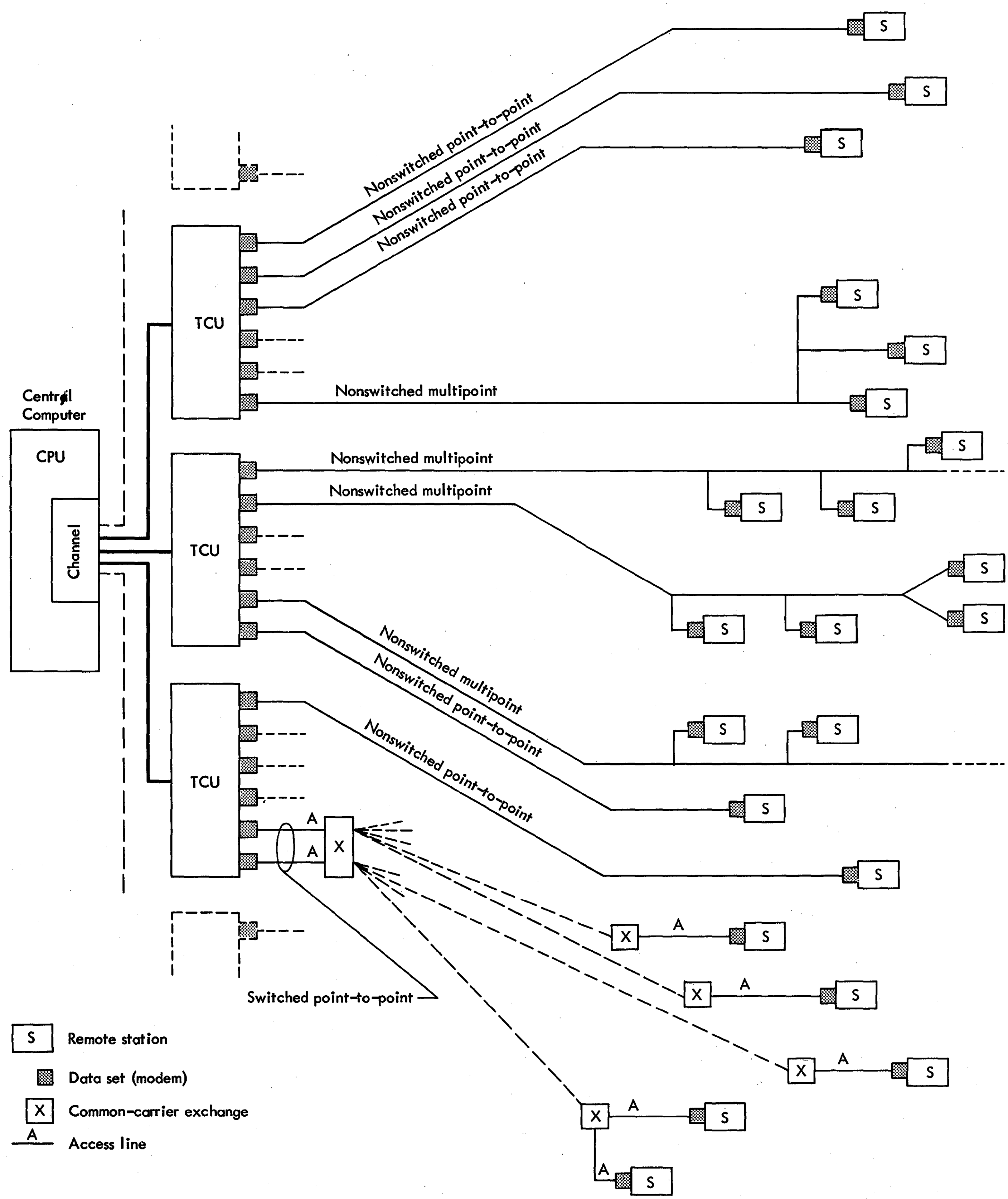


Figure 2. Line and Station Configurations

DUPLEX VS. HALF-DUPLEX TRANSMISSION

The term duplex is applied to a communications line that can accomodate data transmission in both directions at once. Half-duplex lines permit transmission in only one direction at a time. In a BTAM-controlled teleprocessing system, data transmission is always in half-duplex mode; messages are never transmitted in both directions at once.

TRANSMISSION TECHNIQUES

Transmission technique is the way in which data characters are represented on the communications line. The two techniques used by computers and terminals supported by BTAM are start-stop and binary synchronous.

Detailed explanations of these techniques are not given here, as the programmer need not concern himself with them except to specify to BTAM which technique is used. Binary synchronous communication (BSC) is used for high-speed data transmission between the central computer and another remote computer or high-speed terminal. Start-stop transmission (also called asynchronous transmission) is used for data transmission at lower speeds between the central computer and remote terminals of various types.

TRANSMISSION CODES

Data can be represented on a communications line by any of several transmission codes. The code used on a given line is determined by the kind of station or the class of stations connected to the line. Some stations allow a choice of transmission codes. The BTAM programmer must be aware of the code used on a line since he must sometimes specify, in the form of bit patterns, certain data characters to be transmitted by BTAM. At the back of this manual are charts giving the specific bit patterns of the characters contained in the character sets of the various transmission codes or station types.

LINE CONTROL

Just as a computing system, with its variety of peripheral input/output equipment, requires some means to coordinate the functioning of the various parts, the variety of I/O equipment comprising a teleprocessing system requires a discipline to effectively manage the flow of message traffic. A significant difference should be noted, however. In a conventional computing system, the various I/O devices are at the service of the programmer; the requirements of his program and the characteristics of the data to be processed largely determine which input and output devices are to be activated and when. Moreover, the I/O devices are within reach of the computer operator; he can intervene when a device malfunctions to correct the condition or assign a different device. In a teleprocessing system, on the other hand, the central computer receives data at random from remote stations, and the operator at the central computer cannot exercise any direct control over remote stations. He cannot, for example, correct a malfunctioning device at a remote station.

A further distinction between a computing system and a teleprocessing system lies in the handling of errors in data. With current techniques for transmitting data over long distances, errors are frequently introduced into message data by unavoidable transient line conditions such as crosstalk and lightning strikes. Transmission errors occur much less often in a computing system. A discipline for a teleprocessing system must accommodate the facility to detect transmission errors and, when possible, to correct them (as by retransmitting the message containing the errors). If the error is irrecoverable, its occurrence must be signaled to the user program so that appropriate action can be taken.

The scheme of operating procedures and signals by which a teleprocessing system is controlled is called line control (for binary synchronous communications, the term data link control is often used). A line control scheme must consider the functional characteristics and capabilities of the equipment and communication lines composing the system, as well as the operational requirements of the system. Some specific factors that line control must consider are: How is contact to be established between a sending and a receiving station? How is a message to be directed to a specific station on a multistation line? What if two stations try to send at the same time? What should be done if a station fails to respond to a message?

Line control can be classified in two ways. The first way is by the transmission technique (start-stop or binary synchronous) that is used for the line under consideration. With each of these techniques is associated a set of control characters and rules for their use to effect the needed functions. Some of the control characters are used for both start-stop and BSC transmission, while others are peculiar to one or the other of the transmission techniques. The specific line control characters are explained under the discussions of these techniques in the Line Control and Message Transmission chapter.

The second way in which line control can be classified is by the communication line configuration with which it is used. For example, line control for a switched line differs from that for a nonswitched line.

While the general capabilities and functions of a given line control scheme are identified in terms of transmission technique and line configuration, individual variations in capability and function arise from differences in the kind of stations to be controlled, and by the presence or absence in the stations of certain features. For example, a given line control scheme may include the control characters needed to indicate occurrence of a transmission error and to request automatic retransmission, but some types of station equipment that use that line control scheme may not be capable of error checking or automatic retransmission. Generally speaking, all stations connected to a given line must be designed to use the same line control scheme, and where a certain capability is provided by some stations but not by others, the capability cannot be used.

It is not necessary for the BTAM programmer to specify the line control scheme to be used for a given line; this information is provided implicitly at system generation time, and at assembly time in the DCB macro instruction for the line group of which the given line is a member. The programmer must, however, have a general understanding of line control concepts in order to correctly structure that portion of his program that performs message transmission. Also, the programmer must know the meanings of each of the line control characters, as he must regularly insert certain of them into output areas and arrange his program to look for them in input areas.

Line control functions can be considered in two categories: the functions needed to establish contact between central computer and remote stations, and those needed to produce orderly flow of message traffic.

ESTABLISHING CONTACT

Contact may be established in several ways, depending in part upon the line configuration involved.

In some line control schemes one of the stations on a point-to-point nonswitched line can "bid" for use of the line so that it can send a message to the other station. Occasionally both stations may simultaneously bid for use of the line. When this happens, the stations are said to contend with each other; a system in which this situation can occur is called a <u>contention system.</u> The line control scheme for a contention system must provide some means for resolving contention, that is, determining which of the contending stations is to be given the opportunity to send its message. Once one station is given control, the other is blocked from sending. A contention system is more frequently used for a point-to-point line configuration (i.e., involving only two stations) than for a multipoint configuration. BTAM currently provides contention line control only for a point-to-point line.

The alternative to a contention system is a system in which a control station (i.e., the central computer) periodically contacts each of the remote stations in turn and allows it to send any input messages it has ready. ("Ready" means that the terminal operator is prepared to enter data from a keyboard, or that some medium such as cards or paper tape has been placed in an input device so that the data can be transmitted automatically when the control station activates that device.) In this kind of system, each remote station has a unique identifier consisting typically of one or two characters, which, when sent over the line by the control station, causes that remote station, and no other, to respond. In a BTAM-controlled teleprocessing system only the control station, that is, the central computer, activates stations in this manner. The process of contacting in turn each of several stations on a line to determine if any has input ready is called <u>polling</u>, and the station identifiers are called polling characters. Often, the first polling character identifies the station and the second identifies a particular component from which data is solicited. A system in which stations are polled is called a <u>polling system</u> (in contrast to a contention system).

Although the term polling taken in its conceptual sense implies a nonswitched line to which is attached several stations, each of which is solicited in turn, the actual function of polling (that is, sending a station identifier) sometimes applies as well to a point-to-point nonswitched line or to a switched line. In the case of a switched line, the central computer may dial the telephone number of the station (or the station dials the computer) and then the computer transmits the polling characters for that station.

In this discussion of contention versus polling systems, the distinction between the two was based on establishing contact for the purpose of receiving input data from a remote station. The distinction is less clear in the case of output data. In either a contention or a polling system, the central computer must send a station identifier to select the specific station that is to receive an outgoing message. The station identifier in this case is called addressing characters (or selection characters), and the process is called addressing (or selection). As with polling characters, the first addressing character may identify the station and the second, a particular component.

Switched Lines

It should be understood that, in the case of a switched line, the polling and addressing functions are independent of whether the central computer or a remote station initiated the telephone connection. Typically, the operator at a remote station dials the computer only when the remote station has data to send to the computer, and the computer would therefore poll the station after the line connection is established. Similarly, the central computer might dial a remote station only when the computer has data to send, and would therefore address (or select) the remote station. These conventions do not always prevail, however. For example, some applications require that certain stations be polled after working hours when the stations are unattended. With the proper common carrier equipment at the station, the computer can dial the station, then poll the input devices that the operator previously loaded with, for example, a deck of cards or a paper tape.

In establishing contact over a switched line, two situations should be avoided. First, dialing a wrong number can result in establishing contact with a station other than the one intended. Second, an unauthorized station, if provided with the telephone number of the central computer, could establish contact (assuming that the polling or addressing characters corresponded to the characters for authorized stations).

To prevent message transmission under either of these circumstances, identification verification may be used. (This is an optional facility available for certain kinds of stations.) In order to use this facility, each remote station that is permitted to call the computer over a specific switched line termination (i.e., by calling a specific telephone number), must have an identification sequence that it automatically sends after the line connection has been established. The program compares the received sequence against a programmer-defined sequence. If they match, message transmission can proceed; if they differ, BTAM signals the fact by setting a flag bit, and inhibits message transmission. The user's program must check the flag bit and take appropriate action, which ordinarily will be to break the line connection.

If the remote station is a computer, the identification sequence is provided by the programmer, and each computer, central and remote, can check the identity of the other. If the remote station is a terminal, the sequence is mechanically or electrically established when the terminal is installed, and only the central computer can perform the checking function.

Because the central computer has no way of uniquely identifying a station that calls it, all polling and addressing characters and identification sequences must be the same for any station that is to be permitted to call in over a given switched line termination.

The function of identification verification is not applicable to nonswitched lines, since the user determines, when the TP system is installed, which stations are to be connected to a specific nonswitched line.

Terminal Lists

When establishing contact with a remote station, the BTAM program gets the telephone numbers, polling or addressing characters, and identification sequences needed from a control table called a terminal list, which the programmer generates at assembly time using a BTAM macro instruction provided for this purpose. The structure and contents of the terminal list vary according to the kind of line configuration and remote station for which the list is being generated. (Terminal lists are not used for contention systems.)

Positive and Negative Responses

The discussion on how contact is established between stations has up to this point considered only the action taken by the originating station (i.e.,the station that initiates the contact). Before message transmission can proceed, the responding station (the station being contacted) must indicate to the originating station whether or not it is ready to receive or send a message. This indication is generally called a response or answerback, and is termed positive if the station is ready, negative if it is not ready. The specific characters used for positive and negative responses vary with the type of station and the kind of line control (start-stop or binary synchronous) under consideration.

BUFFERING

Buffering is a data management technique often used in conventional (non-teleprocessing) applications because, by permitting greater utilization of input/output areas it minimizes the amount of main storage needed for these areas. This advantage is even more evident in a teleprocessing system, especially one involving many communication lines and varying message lengths.

Each Read or Write operation that involves transfer of text data between a central computer and a communication line requires that an input or output area be assigned to that line. However, to permanently assign main storage areas to each communication line is wasteful, because these areas are idle except during the relative small proportion of time that text transfer to or from the communication line is in progress.

Because data transfer operations are virtually never in progress simultaneously on more than a small proportion of the lines in a system, only a relatively small number of main storage areas are needed to service many communication lines. Buffering permits these areas to be shared among the lines.

Buffering involves defining a group (or pool) of main storage areas, called buffers; assigning buffers from this pool to Read and Write operations as needed; and then returning them to the pool when they are no longer needed, so they may be used for subsequent Read or Write operations.

When the buffer pool is formed, all buffers are chained together by placing a link field containing the address of the next buffer in the chain in the first fullword of each buffer. The link field of the last buffer contains zeros. Adjacent buffers in a chain are not necessarily in contiguous storage locations.

Buffers can be withdrawn from the pool singly or in chains. A buffer control block (BCB) associated with the pool always contains the address of the first available buffer of those remaining in the pool. When buffers are returned to the pool they are automatically inserted into the chain.

A control block associated with each Read and Write operation contains the address of the first buffer in the buffer chain that is assigned to the operation, so the programmer can always determine the address where the received data begins, or where the data to be sent must be placed.

BTAM and the operating system automatically perform the functions necessary to set up a buffer pool when the programmer provides certain information such as the number of buffers he requires and the length of each. (All buffers in a pool have the same length). In addition, BTAM can automatically obtain buffers from the pool and provide them to the Read or Write operation; this is called dynamic buffer allocation, or <u>dynamic buffering</u>. If the programmer does not specify the use of dynamic buffering, the program must request the required number of buffers before initiating the Read operation. This is called <u>programmer buffering</u>.

DYNAMIC BUFFERING

As mentioned earlier, buffering in general provides a significant increase in main storage utilization; dynamic buffering further increases the utilization. With programmer buffering, the programmer must anticipate the length of the message to be received; if messages can be of different lengths, he must request enough buffers to accommodate the longest message that can be expected, even if messages of this length are infrequently received. Furthermore, all buffers are provided in advance of the Read operation, even though they will not all be used at once.

When dynamic buffering is used, however, buffers are obtained singly as the Read operation progresses (by means of program controlled interrupts), and only as many buffers as needed are obtained. When BTAM detects an ending character in a buffer, it does not get any more buffers. Besides allowing delayed acquisition of buffers, dynamic buffering allows buffers that are no longer needed to be progressively released to the buffer pool, instead of remaining idle until the end of the Read or Write operation, and then being released as a group. This technique is possible because, with dynamic buffering, BTAM sets a completion code in the high-order byte of each buffer when the Read or Write operation has finished filling or emptying the buffer. This completion code is of the same kind that is set in the event control block (ECB) at the end of the entire Read or Write operation. The program can check each buffer in turn for this completion code and release the buffer when the code is set.

Whether or not dynamic buffering is employed, BTAM does not release buffers that contain data. The programmer must do this himself, and failure to do so will result in exhaustion of the buffer supply. Should this occur, no more Read or Write operations could be performed.

To summarize, dynamic buffering maximizes buffer utilization by (1) obtaining only as many buffers as are needed for an operation, (2) obtaining them just before actual use, and (3) allowing the programmer to release them, one at a time, immediately after use.

Further information on buffering will be found in the chapter on Buffer Management.

CONVENTIONS USED IN THIS PUBLICATION

To explain the use of BTAM, this publication must frequently express functional relationships between different parts of a teleprocessing system, at varying levels of detail. To express these relationships clearly and concisely requires that certain conventions be observed.

Station, Computer, Terminal: The term station, when not qualified, refers to any of the computers and terminals, whether central or remote, connected to a BTAM-controlled communication line. Central computer means the computer in which the user program under consideration is running. Where the unqualified word computer appears, it means the central computer.

The general term remote station denotes a computer or terminal being controlled by the central computer. Where the context is appropriate, the specific term remote computer or remote terminal is used. For example, in discussions limited to start-stop communication lines, the phrase remote terminal is generally used since start-stop lines do not accommodate remote computers. In discussions of binary synchronous lines, however, the more general phrase remote station is used since BSC lines accommodate both computers and terminals.

Direction of Transmission: The terms input and output are always used relative to the computer in which the BTAM program under consideration is being run. Thus, whether BTAM is running in the central computer or a remote computer, input denotes data transmission from the remote station, and output denotes data transmission from the central computer.

In expressing a specific direction of transmission, the sending and receiving stations are always identified: as in "transmission from central computer to a terminal." The phrase "transmission between central computer and terminal," on the other hand, implies transmission in either direction.

Data, Messages, Text, Control Characters: The term data is the most general of these terms; with respect to communication lines, it refers to any sequence of transmission code bit patterns, whether the patterns represent graphic characters, control characters, or binary information. Message means any sequence of data characters, considered as a unit, and includes any control characters necessary for transmission on a communication line. Text refers to the data characters comprising the information to be conveyed, such as plain language or binary data. Control characters are characters needed either to control transmission on the line (called line control or data link control characters) or to activate mechanical or formatting functions at a station (end-to-end control characters). Examples of line control characters are SOH, STX, and EOT (start of heading, start of text, end of transmission). Examples of end-to-end control characters are CR, LF, VT, and BEL (Carriage Return, Line Feed, Vertical Tab, Bell).

Usually, the name of a character and the function it performs are the same, e.g., an EOT character indicates the end of transmission. In some cases, however, a particular function is effected by a different character or character sequence. For example, the EOA (end-of-address) character is sometimes used as a positive response signal, and for certain non-IBM terminals the characters FIGS H LTRS are employed as an end-of-transmission sequence. Where these disparities of function and character name occur, the intended meaning is made clear.

The text portion of an output message is given by the user to BTAM in a work area or buffer. The user also must provide certain line control characters in the buffer.

Read and Write Operations: The sequence of events by which data characters are sent or received is called a Read operation for input messages, and a Write operation for output messages.

Each Read or Write operation is produced by a READ or WRITE macro instruction issued by the user's program (except for some operations performed automatically by error recovery procedures and on-line testing facilities). The term Read (or Write) operation may be qualified at several levels. For example, the phrase "Read operation" refers to any of several types of Read operation; the phrase "Read Initial operation" refers to any of several variations of Read Initial operations, and so on. Where a specific type is intended, the corresponding type code is usually given, as in "Read Initial Conversational (TIV) operation."

A Read or Write operation is performed by a sequence of commands executed by the channel to which the transmission control unit (TCU) is connected. These channel commands cause the TCU to transmit data characters and control signals on the line or, conversely, to respond to data characters and signals received from the line.

In discussions of Read and Write operations, the term command means a channel command, as represented in main storage by a channel command word (CCW).

BTAM macro instructions, like other operating system macros, are written in the assembler language, and accordingly are subject to the rules given in IBM System/360 Operating System: Assembler Language (GC28-6514). BTAM macros, like all assembler language macros, are coded in this format:

Name	Operation	Operands
Symbol or Blank	Macro Name	One or more operands separated by commas.

The operands are used to specify the facilities to be included, services to be performed, and various parameters needed by BTAM. Operands are coded according to the following rules.

Positional Operands

Positional operands are shown as either small letters or capital letters. Small letters describe the kind of information to be coded; capital letters indicate the exact characters to be coded.

If the operand is shown as small letters (e.g., inarea), substitute for it one of the values shown in the macro instruction format chart, Appendix G.

If the operand is shown as capital letters (e.g., OPENLST), code it exactly as shown.

Code commas and parentheses exactly as shown. If an optional operand is omitted a comma must still appear, except that no commas need follow the last operand coded.

Keyword Operands

Keyword operands are shown as a word, in capital letters, followed by an equal sign, followed by (1) a descriptive word or phrase, in small letters, or (2) a specific character or sequence of characters, in capital letters.

If small letters follow the equal sign, code the keyword and equal sign exactly as shown, and substitute for the word or phrase in small letters one of the values shown in the macro instruction format chart.

If a specific character sequence follows the equal sign, code the entire operand exactly as shown.

Code commas and parentheses exactly as shown. Unlike positional operands, no comma need be coded in place of an omitted optional keyword operand.

Continuation Lines

The operand field of a macro instruction can be continued on one or more additional lines as follows:

1. Enter a continuation character (any nonblank character that is not part of the operand coding) in column 72 of the line.

2. Continue the operand field on the next line, starting in column 16. All columns to the left of column 16 must be blank.

The operand field being continued can be coded in one of two ways. You may code the operand field through column 71, with no blanks, and continue the coding in column 16 of the next line, or you may truncate the operand field at the end of an operand (including the comma that follows the operand), then start the next operand in column 16 of the next line.

Examples:

Name	Operation	Operand	Col 72
NAME1	OP1	OPERAND1,OPERAND2,OPE RAND3,OPERAND4,OPERAN D5,OPERAND6 THIS IS ONE WAY	X X
* 			
NAME2	OP2	OPERAND1,OPERAND2, OPERAND3, THIS IS OPERAND4 ANOTHER WAY	X X
*			

Coding Aids

The symbols [ ] (brackets), { } (braces), and ... (ellipsis) are used to aid in defining macro instruction formats; they are never coded. Their meanings are as follows.

[ ]
indicates that the enclosed operand is optional, or is coded under certain conditions. If more than one operand is stacked within brackets, as for example,

$$\begin{bmatrix} \text{CLEAR=NO} \\ \underline{\text{CLEAR=YES}} \end{bmatrix}$$

then one of the items, or none, may be coded. If one of the choices is underlined, the option associated with that choice is assumed to be specified if that operand is not coded.

$\left\{ \right\}$
indicates that one of the enclosed operands must be coded.

{ }
defines the limits of a syntactical unit, where the unit consists of more than one operand, e.g., ({operandone, operandtwo},...). The { } and ellipsis signifies that the two operands may be repeated in sets. The enclosing parentheses must be coded.

Defining a teleprocessing system means specifying to BTAM and the operating system the characteristics of the communications lines, transmission control units, and remote stations comprising the system. Using this information, the operating system establishes the appropriate program interfaces between the user program and data management input/output routines.

At system generation, the programmer must specify the types of transmission control units and remote stations composing the TP system, and the features associated with each TCU and station. Appendix D explains how to code system generation macro instructions to provide this information. (See the System Generation publication for general information on the system generation procedure.)

At assembly time and during program execution, the programmer must define communications line groups and terminal lists, as explained below.

For information about defining the local 3270 display system and about attention interruption handling, see the section "IBM 3270 Display System - Programming Considerations."

DEFINING COMMUNICATIONS LINE GROUPS

A communications line group is a logical association of lines having characteristics similar enough that the same channel programs can be used for all lines in the group. These characteristics are as follows:

- All lines in a group must be start-stop, or all must be binary synchronous; both types cannot be mixed in the same group.
- All lines must be of the same type. For start-stop lines, this means that all lines in the group must be nonswitched, or all must be switched. For binary synchronous lines, all lines in the group must be nonswitched point-to-point, nonswitched multipoint, or switched point-to-point.
- All remote terminals connected to start-stop lines in a group must be of the same type, must have the same features, and must use the same transmission code.
- All remote stations connected to binary synchronous (BSC) lines in a line group must have the same features and must use the same transmission code, but they need not all be of the same type. That is, more than one type (e.g., System/360 Model 20, 1130, 2770, etc.) can be connected to the same nonswitched multipoint line, or more than one type can call or be called by the central computer over the same switched line termination (i.e., the same telephone number at the central computer).
- Any optional functions that are specified apply to all lines in the group. For example, if dynamic buffering is specified for the line group, all lines must use dynamic buffering.

Communications line groups are considered as data sets, although they do not conform to the usual definition of data set (a named, organized collection of logically related records). Like other data sets, a communications line group is represented by a data control block (DCB), which you define using the DCB macro instruction.

You may establish a line group in either of two ways. First, you may specify during system generation what lines are to constitute the group, then symbolically identify the group during program execution by means of the UNIT parameter of a DD statement. Second, you may specify the makeup of the group during program execution, again using UNIT parameters on DD cards. Whichever method is used, during system generation you must identify to the operating system (1) what kind of transmission control unit is connected to each line, and (2) what kind of terminals (start-stop) or what type of line (BSC) is associated with each line address.

A series of examples will make this clear. Assume that the teleprocessing network comprises seven lines -- five start-stop and two binary synchronous (BSC) lines. Assume further that IBM 1050 terminals are connected to the five start-stop lines, and that any type of BTAM-supported BSC station is connected to the two BSC lines. See Figure 3. (The decision as to what types of remote stations are to be connected to which lines is part of the installation planning function; this example assumes that this decision has already been made.)

Before establishing the makeup of a line group, you must specify with IOCONTRL macros the type of TCU (2701, 2702, or 2703) associated with each control unit address, and then, via IODEVICE macros, specify the type of terminal (for start-stop lines) or type of line (for BSC lines) associated with each line address.

```
IOCONTRL  UNIT=2702,ADDRESS=02
IOCONTRL  UNIT=2703,ADDRESS=03
```

The addresses 02 and 03 represent the control unit addresses to which the TCUs are connected. Note that one IOCONTRL macro is issued for each control unit position (denoted by the second digit of the line address).

```
IODEVICE  UNIT=1050,ADDRESS=021,...
IODEVICE  UNIT=1050,ADDRESS=022,...
IODEVICE  UNIT=1050,ADDRESS=025,...
IODEVICE  UNIT=1050,ADDRESS=027,...
IODEVICE  UNIT=1050,ADDRESS=02E,...
IODEVICE  UNIT=BSC3,ADDRESS=031,...
IODEVICE  UNIT=BSC3,ADDRESS=033,...
```

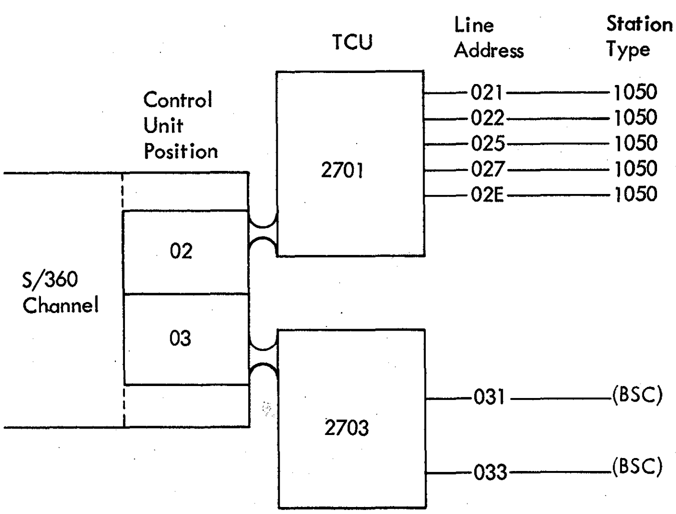


Figure 3. Sample Line Addresses and Associated TCU and Station Types

Notice that the UNIT operands in the first five macros specify the type of terminal, while those in the remaining two macros specify "BSC3" -- this value represents the type of line, in this case non-switched multipoint. To specify a non-switched point-to-point line you would specify BSC1; for a switched point-to-point line, BSC2. See the description of the IODEVICE macro in Appendix D.

In these macros, the three-digit addresses represent communications lines, not specific devices as would be the case in defining local I/O equipment. The ellipses represent other appropriate IODEVICE operands, including FEATURE, ADAPTER, and SETADDR. These are explained in Appendix D.

Assume now that you wish to establish three line groups from the seven lines, as shown in Figure 4. Two examples show how to do this.

Example 1: To define the groups at system generation, you would code a UNITNAME macro for each line group:

```
UNITNAME UNIT=(021,022,025,02E),NAME=GROUP1
UNITNAME UNIT=(027),NAME=GROUP2
UNITNAME UNIT=(031,033),NAME=GROUP3
```

During program execution, you would associate these line groups with specific data control blocks in your program by issuing for each group a DD statement identifying the group by its name, and indicating the number of lines in the group:

```
//DDGRP1 DD        UNIT=(GROUP1,4),...
//DDGRP2 DD        UNIT=(GROUP2,1),...
//DDGRP3 DD        UNIT=(GROUP3,2),...
```

The name of the DD statement (e.g., DDGRP1) must be the same as the DDNAME parameter in the data control block. In the foregoing DD statements the number of lines indicated for each group is the same as the number of lines specified at system generation in the UNITNAME macros. You may, however, specify any lesser number of lines to be in the group when you issue a DD statement. Lines are always included beginning with the lowest line address.

For example, if you code

```
//DDGRP1 DD        UNIT=(GROUP1,2),...
```

the two lines associated with the two lowest addresses in the original four-line group, 021 and 022, constitute the group.

Example 2: Alternatively, you may elect not to define the groups at system generation, but to identify the lines in the group individually during program execution. To do this, code a separate DD statement for each line, specifying the actual line address instead of the name of a line group, as in example 1.

```
//DDGRP1 DD        UNIT=021,...
//       DD        UNIT=02E,...
//       DD        UNIT=027,...
```

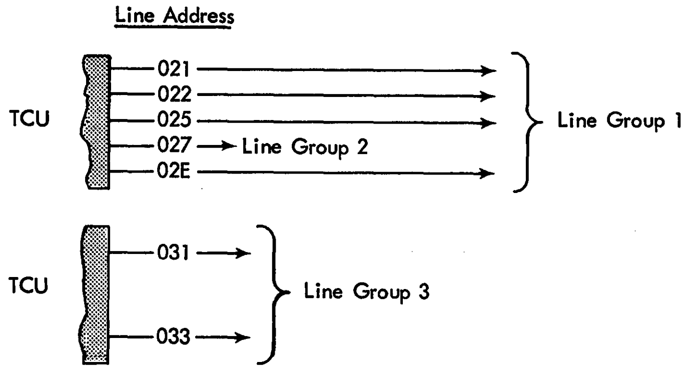


Figure 4. Sample Line Group Makeup

These DD statements establish a group containing three lines.

Even if you have defined a line group by using a UNITNAME macro at system generation you may redefine it by using DD statements as shown in this example.

<u>Relative Line Numbers</u>: In READ and WRITE macro instructions, you must specify the line over which the Read or Write operation is to take place. Rather than specify an actual line address, you code in the macro a <u>relative line number</u> (RLN), which refers to the numerical position of the actual line address relative to other lines in the group. Thus, in Example 1, the relative line numbers are as shown in Figure 5; in Example 2, as shown in Figure 6. (In Example 2 four lines are not defined in any group.)

If you define a line group at system generation, using the UNITNAME macro, the relative line numbers are assigned such that they represent an ascending numerical order of addresses, regardless of the actual sequence in which the addresses were arranged in the macro. If, however, you define a line group by a sequence of DD statements, relative line numbers are applied in the same sequence in which the DD statements are coded.

See Appendix D for further information on system generation macro instructions involved in generating a BTAM system, and see the <u>System Generation</u> publication for complete information on generation procedures.

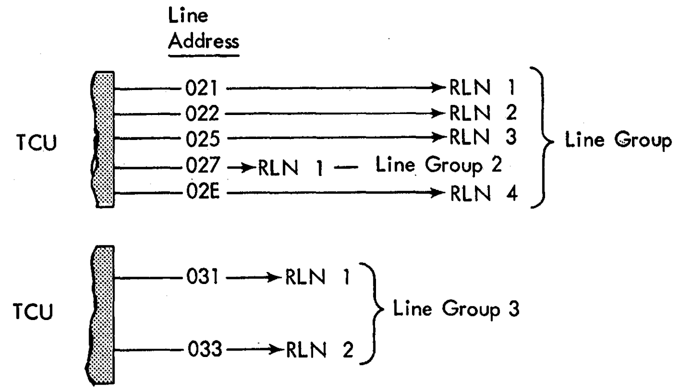


Figure 5. Relative Line Numbers for Example 1

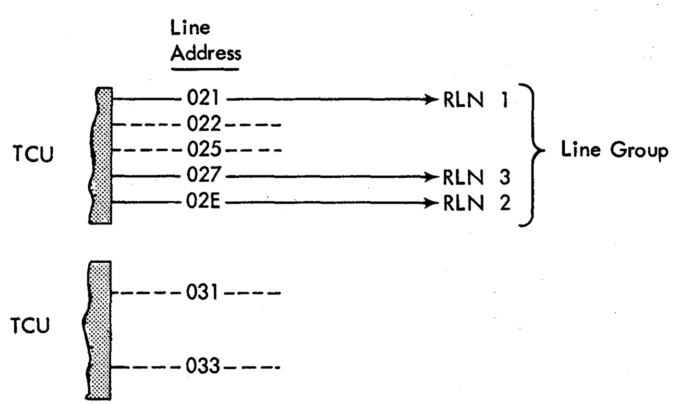


Figure 6. Relative Line Numbers for Example 2

<u>DCB (Define Data Control Block) Macro Instruction</u>

The DCB macro instruction defines the structure of a data control block and includes in it certain information that you have coded in the macro. You must issue a separate DCB macro for each line group data set.

Of the parameters that appear in the data control block, you must code certain ones in the macro, while others may either be coded in the macro or be supplied from an alternate source. The alternate source for an operand is indicated in the description of the operand, as follows:

- PP means you can enter the parameter into the data control block yourself during program execution, any time prior to opening the line group data set.

- OE means you can enter the parameter into the data control block yourself during program execution, at any time up to and including the DCB exit taken during the opening process.

Name	Operation	Operands
symbol	DCB	keyword operands

symbol
: Is the name of the DCB macro instruction. It must be specified.

keyword operands
: Are the operands that can be included (Table 1).

Table 1. Keyword Operands for the BTAM Communications-Line-Group DCB Macro Instruction (Part 1 of 5)

Keyword Operand and Description
DSORG=CX Identifies the data set organization as that of a communications line group.
{MACRF=(R) / MACRF=(W) / MACRF=(R,W)} Specifies that access to the line group is to be gained with either READ or WRITE macro instructions, or both. Whichever option is coded, BTAM permits access with both READ and WRITE macro instructions. This operand is required.
[DDNAME=ddname] (Alternate source: PP) Is the name that appears in the DD statements associated with this data control block. If this operand is omitted, and no value is provided through an alternate source, the job is terminated.
[BUFNO=number of buffers] (Alternate source: OE) Is the number of buffers to be obtained by BTAM at open time, if you wish BTAM to provide a buffer pool. Up to 225 buffers can be specified. You need not code this operand if BTAM is not to obtain a buffer pool.
[BUFL=buffer length] (Alternate source: OE) Is the length in bytes of the buffers making up the buffer pool, whether you provide the pool or BTAM provides it. The maximum value for BUFL is 32,760. A minimum limit on buffer length applies to BSC line groups under certain conditions -- see Programming Notes under READ and WRITE Macro Instructions. Specify this operand for all applications using buffers. BUFL must be a multiple of 4.
[BUFCB=buffer control block address] (Alternate source: OE) Specifies the address of the buffer control block for a buffer pool you provide. If you wish BTAM to provide the buffer pool, omit this operand, and code the BUFNO and BUFL operands.
[EXLST=exit list address] (Alternate source: PP) Specifies the address of a BTAM program exit list, if you wish to provide one. Only the DCB exit may be used.
[BFTEK=D] (Alternate source: OE) Specifies dynamic buffering is to be used for this line group. If dynamic buffering is specified, a buffer pool must be defined.
[LERB=line error block address] (Alternate source: OE) Specifies address of line error recording block. This operand is valid only if C is coded among the EROPT operand options.
[EROPT=code] (Alternate source: OE) Specifies the error recovery, error recording, and on-line test options to be provided for the line group. E Specifies that the basic error recovery procedures (ERP) are to be provided for the line group. If EROPT is omitted, E is assumed. R Specifies that text-read errors are to be retried in addition to the basic error recovery procedures. This option is valid only for the following terminals:

Table 1. Keyword Operands for the BTAM Communications-Line-Group DCB Macro Instruction (Part 2 of 5)

Keyword Operand and Description
1050 terminals (valid for the card reader and paper tape reader only if line correction feature is installed), 2740 terminals with checking feature, and 2260 terminals. (Do not specify EROPT=R if dynamic buffering is to be used (BFTEK=D) as the use of dynamic buffering precludes the retrying of text-read errors. (See also discussion under N, below, for considerations for AT&T 83B3 and WU 115A terminals.)
W
Specifies text-write errors are to be retried in addition to basic error recovery procedures. This option is valid for all start-stop terminals, except World Trade terminals. It is invalid for BSC stations. It results in an additional copy of the message for each retry (except for the 2260 with the line address feature, and the 1050 card punch and paper tape punch with the line correction feature). This parameter is ignored for BSC and World Trade telegraph terminals. (Do not specify EROPT=W if dynamic buffering is to be used (BFTEK=D) as the use of dynamic buffering precludes the retrying of text-write errors.
C
Specifies that threshold error counts and cumulative error counts are to be maintained in the line error recording block (LERB) for the line for data check, intervention required, and nontext time-out errors.
N
Specifies that no error recovery procedures are to be provided for the line group. This parameter and E,R,W, and C are mutually exclusive. This parameter is invalid for BSC stations; if coded, it is ignored. It is recommended that EROPT=N for AT&T 83B3 and WU 115A terminals if dynamic buffering is specified (BFTEK=D), because BTAM does not perform error retry either before or after start of text transfer when dynamic buffering is used for terminals of these types. If EROPT is omitted, or E, ER, or R is coded in the EROPT operand, ERP routines are unnecessarily loaded into the system, as they will remain unused.
T
Specifies that the on-line test facility is to be used for the line group. This option is valid for all IBM stations with or without error recovery procedures. To receive standard IBM maintenance for a remote or local 3270 display system, this option must be specified.
<u>Note:</u> The parameters E, R, W, C, and T may appear in any combination. The parameter N may appear alone or with T. Commas must not be coded in this parameter. Example: EROPT=RECWT. When EROPT (any combination of E, R, W, and C) is coded in the DCB macro instruction, the user automatically gets the Outboard Recorder (OBR) and the Statistical Data Recorder (SDR) facilities for this line group. (These are facilities used by the Customer Engineer.) Error recovery procedures are required for BSC stations. For BSC line group data sets C and T are the only valid EROPT options; all other option codes are ignored. For World Trade terminals, E, C, and N are the only valid EROPT options; all other option codes are ignored. For the local 3270 display system, E and T are the only valid EROPT options; all other option codes are ignored.
<u>Caution:</u> In previous releases of the S/360 Operating System, the EROPT operand of the BTAM DCB macro was spelled ERROPT. In a user program assembled under the current release of the Operating System, this operand must be spelled EROPT. The assembler will issue an MNOTE for, and will not assemble, any BTAM DCB macro in which the operand is coded ERROPT.

Table 1. Keyword Operands for the BTAM Communications-Line-Group DCB Macro Instruction (Part 3 of 5)

Keyword Operand
$\begin{bmatrix}\text{DEVD=BS}\\\text{DEVD=WT}\end{bmatrix}$ BS Specifies that BSC is to be used and causes a 44 byte field to be added to the DCB. This field contains the line control characters in the transmission code to be used. WT Must be coded if the line group contains World Trade terminals, and if any of the keyword operands IAM,WRU,MON,MONDLY,EOM, and EOT are coded.
[MODE=([IBC],[CNTRL],$\begin{bmatrix}\underline{\text{A}}\\\text{B}\end{bmatrix}$,$\begin{bmatrix}\underline{\text{A}}\\\text{B}\end{bmatrix}$)] (BSC line group only) IBC Specifies that the transmission control unit (TCU) at the central computer is to operate in EIB (Error Information Byte) mode. EIB mode is discussed in the General Information section of the BSC Read and Write Operations chapter. CNTRL Should be coded if the central computer (this System/360) is to be given control when contention occurs on a point-to-point non-switched line. It should be omitted if the remote station is to be given control. <u>A</u> Specifies that communications are to be through the 2701 Data Adapter Unit's Dual Communication Interface A. B Specifies that communications are to be through the 2701's Dual Communication Interface B. This parameter may not be coded if this feature is not present on the 2701. <u>A</u> Specifies use of the transmission code designated by Code A for 2701 Data Adapter Unit Dual Code Feature. B Specifies use of the transmission code designated by Code B for 2701 Dual Code Feature. This parameter may not be coded if this feature is not present on the 2701.
[CODE=transmission code] (BSC line group only) <u>EBCDIC</u> Specifies transmission in Extended Binary Coded Decimal Interchange Code. USASCII Specifies transmission in United States of America Standard Code for Information Interchange. TRANSC Specifies transmission in 6-bit TRANSCODE.

Table 1. Keyword Operands for the BTAM Communications-Line-Group DCB Macro Instruction (Part 4 of 5)

Keyword Operand and Description
The following six operands apply only to line groups for World Trade Telegraph terminals:
[MON=YES / MON=NO]   YES   Specifies that each terminal of the line group is equipped with the optional Motor-On feature.   NO   Specifies that the terminals are not equipped with the Motor-On feature. NO is assumed if this operand is omitted.
[MONDLY=nn / MONDLY=15]   nn   Specifies the number of Mark Characters corresponding to a 1.5-second time-out when the terminal is not equipped with the optional Motor-On feature. MONDLY=10 corresponds to 50-baud service, MONDLY=15 corresponds to 75-baud service, and MONDLY=20 corresponds to 100-baud service. When this operand is omitted or nn exceeds 20, MONDLY=15 is assumed.
[IAM=YES / IAM=NO]   YES   Specifies that the terminal can ask for the computer identification sequence by sending FIGS D.   NO   Specifies that the terminal cannot ask for the identification sequence of the computer. NO is assumed if this operand is omitted.
[WRU=YES / WRU=NO]   YES   Specifies that by sending FIGS D, either the computer or the terminal can ask for the identification sequence of the other. When WRU=YES is specified, IAM=YES is assumed.   NO   Specifies that the computer cannot ask for the identification sequence of the terminal. NO is assumed if this operand is omitted.

Table 1. Keyword Operands for the BTAM Communications-Line-Group DCB Macro Instruction (Part 5 of 5)

Keyword Operand and Description
[EOM=WRU / EOM=X'hh' / EOM=X'hhlF'] (EOM=WRU underlined as default) WRU Specifies that the end-of-message signal is the WRU signal. X'hh' Specifies that FIGS x is used as the EOM signal.[1] hh is the hexadecimal representation of FIGS x set in the adapter. X'hhlF' Specifies that FIGS y LTRS is used as the EOM signal.[1] hh is the hexadecimal representation of FIGS y set in the adapter. WRU is assumed if this operand is omitted.
[EOT=2EOM / EOT=X'hhlF'] (EOT=2EOM underlined as default) 2EOM Specifies that two consecutive EOM signals will be recognized by BTAM as end-of-transmission, except when IAM=YES and EOM=WRU are specified. X'hhlF' Specifies that FIGS y LTRS is used as the EOT signal.[1] Therefore, EOM=X'hhlF' cannot be specified for the EOM signal. Note: A time-out is also recognized as EOT. Moreover, two consecutive EOM signals are always recognized as an EOT signal, except when IAM=YES and EOM=WRU are specified.
[1]x and y are the values assigned by the user and set in the adapter at the time of installation of the equipment.

Table 2. Format of Data Control Block (DCB) (Part 1 of 2)

Displacement Hex	Dec					
10	16	DCBBQFLG	DCBWTEOM	DCBWTEOT	DCBWTPAD	World Trade Telegraph Interface (before and after the Open Function)
14	20	DCBBUFNO	DCBBUFCB			Common Interface
18	24	DCBBUFL		DCBDSORG		
1C	28	DCBDEVTP	DCBIOBAD			
20	32	DCBBFTEK	DCBERROP	DCBBUFCT		Foundation Extension
24	36	DCBEIOBX	DCBEXLST			
28	40	DCBDDNAM				Foundation Before Open
2C	44					
30	48	DCBOFLGS	DCBIFLG	DCBMACR		
28	40	DCBTIOT		DCBMACRF		Foundation After Open
2C	44	DCBIFLGS	DCBDEBAD			
30	48	DCBOFLGS				
30	48		DCBREAD,DCBWRITE			BTAM Interface
34	52	DCBLERB				
38	56		DCBXCODE			BSC Interface (before Open)
3C	60	DCBBSTSX				
40	64	(reserved)				

Table 2. Format of Data Control Block (DCB) (Part 2 of 2)

38	56	DCBXMODE	DCBXCODE	DCBBSRSV	DCBBSWBT	BSC Interface (after Open)
3C	60	DCBBSTSX	DCBBSSTX	DCBBSTEX	DCBBSETX	
40	64	DCBBSAK0		DCBBSAK1		
44	68	DCBBSENQ	DCBBSNAK	DCBBSETB	DCBBSDLE	
48	72	DCBBSEOT	DCBBSSYN	DCBBSTBE	DCBBSTEB	
4C	76	DCBBSONL		DCBBSSAK		
50	80	DCBBSRVI		(reserved)		
54	84	(reserved)				
60	96					

Table 3. DCB Field Contents

Field	Contents	
DCBBQFLG	World Trade Telegraph flag byte.	
DCBWTEOM	The EOM character (WT terminals)	
DCBWTEOT	The EOT character (WT terminals)	
DCBTPAD	Number of pad (LTRS) characters required for Motor-on delay (WT terminals).	
DCBBUFNO	Number of buffers, obtained by Open-routine for this DCB.	
DCBBUFCB	Address of buffer control block.	
DCBBUFL	Buffer length (length of buffers to be obtained by Open for a BTAM-provided buffer pool, and/or the buffer length to be used if length parameter of a Read or Write macro is coded as 'S').	
DCBDSORG	Data set organization (bit 3=1 for BTAM)	
DCBDEVTP	Index to device entry in Device I/O directory.	
DCBIOBAD	IOB (Input/Output Block) address.	
DCBBFTEK	Buffering technique, (Bit 4=1 indicates dynamic buffering)	
DCBERROP	Error recovery procedures defined by DCB EROPT operand.	
DCBBUFCT	Max. no. of buffers to be obtained by BTAM for a Read or Write operation (dynamic buffering).	
DCBEIOBX	Extended IOB index.	
DCBEXLST	Address of a user-provided exit list.	
DCBDDNAM	DD name of the line group data set.	
DCBOFLGS	Flags used by OPEN, and checked by programmer to determine if data set has been opened.	
DCBIFLG	Flags used by Input/Output Supervisor (IOS).	
DCBMACR	Macro instruction reference.	
DCBTIOT	Pointer to DD entry in task I/O table.	
DCBMACRF	Same as DCBMACR.	
DCBIFLGS	Same as DCBIFLG.	
DCBDEBAD	Address of associated Data Extent Block (DEB).	
DCBOFLGS	Same as DCBOFLGS above.	
DCBREAD/DCBWRITE	Address of Read/Write module.	
DCBLERB	Address of line error recording block (LERB).	
DCBXMODE	Transmission mode for BSC lines.	
DCBXCODE	Transmission code for BSC lines.	
DCBBSRSV	The DLE character.	Hexadecimal representation of transmission code
DCBBSWBT	(reserved)	
DCBBSTSX	DLE character.	
DCBBSSTX	STX character.	
DCBBSTEX	DLE character.	
DCBBSETX	ETX character.	
DCBBSAK0	ACK-0 sequence[1].	
DCBBSAK1	ACK-1 sequence[2].	
DCBBSENQ	ENQ character.	
DCBBSNAK	NAK character.	
DCBBSETB	ETB character.	
DCBBSDLE	DLE character.	
DCBBSEOT	EOT character.	
DCBSSYN	SYN character.	
DCBBSTBE	DLE character.	
DCBBSTEB	ETB character.	
DCBBSONL	SOH % characters.	
DCBBSSAK	WACK sequence[3].	
DCBBSRVI	RVI sequence[4].	

[1]ACK-0 is two characters -- DLE X'70' (EBCDIC), DLE 0 (USASCII), or DLE - (TRANSCODE).

[2]ACK-1 is two characters -- DLE / (EBCDIC), DLE 1 (USASCII), or DLE T (TRANSCODE).

[3]WACK is two characters -- X'106B' (EBCDIC), X'103B' (USASCII)

[4]RVI is two characters -- X'107C' (EBCDIC), X'103C' (USASCII)

DEFINING AND MODIFYING TERMINAL LISTS

A terminal list is a table from which BTAM obtains the information it needs to establish contact with a remote station when you issue a READ Initial or WRITE Initial macro instruction (and occasionally other types of READ and WRITE). This information consists of telephone numbers (dial digits), polling and addressing sequences, and identification sequences to be sent to remote stations, or against which an incoming sequence can be checked to ensure that contact has been established with a valid station.

There are several kinds of terminal lists, having different names and formats. For example, a polling list is one kind of terminal list; it is used for supplying the polling sequences BTAM needs to activate certain kinds of remote stations. Another kind is a dial list, used in operations over switched lines. Appendix A shows the formats of terminal lists and gives examples of what they contain.

Two macro instructions, DFTRMLST and CHGNTRY, provide the ability to define terminal lists and to modify an existing list.

DFTRMLST (Define Terminal List) Macro Instruction

DFTRMLST generates a terminal list having the format and contents required by the type of station and kind of communication line involved in the Read or Write operation that uses the list. The macro specifies the format and provides telephone numbers, polling or addressing characters, and identification sequences, as required by the Read or Write operation.

Described below are each of the operands that may be coded in a DFTRMLST macro instruction; only a few of these will be coded for a particular list. To determine which ones to code for a particular Read or Write operation, see the discussion on terminal lists in the section of the Start-Stop Read and Write Operations or BSC Read and Write Operations chapters that covers the particular type of remote station or line configuration for which the terminal list is required.

A separate DFTRMLST macro must be issued for each list to be defined. Appendix A illustrates the formats of various kinds of lists, with examples.

Notes: The DFTRMLST macro instruction is not used for the local 3270 display system.

For more information about the DFTRMLST macro instruction for the remote 3270 display system, see "Defining Terminal Lists" under the heading "Defining and Modifying Terminal Lists" in the section "IBM 3270 Display System - Programming Considerations."

Name	Operation	Operand
symbol	DFTRMLST	list type, device-dependent operands

list type
: Specifies the format of the terminal list. Code one of the following, as required by the remote station or line configuration involved.

OPENLST (start-stop, BSC; multipoint line)
: Generates an open polling list (for programmed polling of start-stop terminals) or an addressing list (for addressing of start-stop or BSC stations).

WRAPLST (start-stop; multipoint line)
: Generates a wraparound polling list (for programmed polling).

SSALST (start-stop; multipoint line)
: Generates an open polling list for Auto Poll operations.

SSAWLST (start-stop; multipoint line)
: Generates a wraparound polling list for Auto Poll operations.

DIALST (start-stop, BSC; switched line)
: Generates a calling list or an answering list.

IDLST (start-stop [TWX only]; switched line)
: Generates a calling list or an answering list with ID verification.

BSCLST (BSC; switched line)
: Generates a calling or an answering list (for switched lines), with ID verification. This kind of list specifies an identification sequence to be sent to a remote BSC station; and specifies what identification sequence will be accepted from a remote BSC station.

AUTOLST (BSC; multipoint line)
: Generates an open polling list for Auto Poll operations.

AUTOWLST (BSC; multipoint line)
Generates a wraparound polling list for Auto Poll operations.

WTTALST (start-stop [WT telegraph only]; nonswitched point-to-point line)
World Trade telegraph terminal list containing the identification sequence expected from a remote station and the identification to be sent to the remote station when transmission begins.

WTLIST (BSC; switched line)
Generates a terminal list to be used for Read and Write operations involving manual dialing of a remote station or manual answering of calls from remote stations, where the expanded ID verification facility is not to be used (i.e., when only one unique ID sequence is to be accepted from any remote station that calls or is called by the central computer).

SWLST (BSC; switched line)
Generates a terminal list to be used for Read and Write operations involving automatic or manual dialing of a remote BSC station or automatic answering of calls from remote BSC stations, where the expanded ID verification facility is to be used (i.e., when any of several authorized ID sequences is to accepted from a remote station).

device-dependent operands
Specify the information to be placed in the list.

xx
two hexadecimal digits representing the transmission bit pattern of a single polling or addressing character. Example: 62 (representing the polling character A in transmission code [1030]).

xxyy
four hexadecimal digits representing the transmission code bit patterns of a two-character polling or addressing sequence. Example: E202 (representing the polling characters A1 in transmission code [1050]).

dialcount
one or two decimal digits representing the number of dial digits in the telephone number of the remote station to be called. Example: 7.

dialchars
the digits of the telephone number to be dialed. Example: 5672022.

numrec
one or two decimal digits representing the number of characters in an identification expected from a remote station.

ridseq
hexadecimal digits representing the transmission code bit patterns of the identification sequence to be received.

numsent
one or two decimal digits representing the number of characters in the identification sequence to be sent to a remote station.

tidseq
hexadecimal digits representing the transmission code bit patterns of the identification sequence to be sent.

numcnsent
one or two decimal digits representing the number of characters in a terminal control sequence to be sent to a TWX station.

cntrlseq
hexadecimal digits representing the transmission code bit patterns of the terminal control sequence to be sent.

length
the number of characters composing a "data tone" (an audible signal to be sent to a remote station that calls the central computer). Code this operand only for lists of the WTLIST type. (A sequence of X'FF' characters is recommended for the data tone.)

area
the address of the area containing the data tone character sequence. Code this operand only if you code the length operand.

faaseq (2760 only)
hexadecimal digits representing the transmission code bit patterns of the three-character frame change sequence (F, A_1, A_2 characters).

Programming Note: The DFTRMLST macro cannot define open or wraparound lists of the OPENLST or WRAPLST types having more than 31 entries. If a larger list is required, you must define it yourself; see Appendix A for the required format.

CHGNTRY (Change Terminal Entry) Macro Instruction

CHGNTRY is used to cause BTAM to suspend or resume polling or addressing of a specific remote station or component represented by a terminal list entry or to change the value of a control byte in an answering list of the SWLST form.[1] For a programmed polling list or an addressing list, CHGNTRY sets the skip bit of the entry to 1, if polling or addressing is to be skipped; or sets the bit to 0, if polling or addressing is to be resumed. For an Auto Poll polling list, CHGNTRY moves an entry to be skipped to the end of the list so that all active entries appear at the beginning of the list, and all entries to be skipped appear at the end of the list. CHGNTRY moves an entry to be reactivated back to its original position in the list.

You must issue a separate CHGNTRY for each list entry you wish to skip or activate, or for each SWLST control byte value to be changed.

You can change a terminal list entry only if the list is not currently in use by a Read or Write operation. You should therefore issue CHGNTRY only after making sure that no Read or Write operation is in progress on the line to which the list applies. If you wish to change the list while wraparound polling is in progress, first issue a RESETPL macro to terminate polling, then issue a CHGNTRY for the entry to be changed.

CHGNTRY cannot be used to modify a terminal list of the IDLST or BSCLST format.

Note: A special form of the CHGNTRY macro instruction is used for the local 3270 display system. See "Attention Interruptions and Read Initial Operations" in the section "IBM 3270 Display System—Programming Considerations."

Name	Operation	Operand
[symbol]	CHGNTRY	listaddr,listype, listposition,numchars, action

----

[1] CHGNTRY cannot be used to change the control byte value of a calling list of the SWLST form.

listaddr
: Specifies the address of the first entry of the terminal list containing (1) the entry to be skipped or activated, or (2) the SWLST entry the control byte value of which is to be changed.

listype
: Specifies the type of list, as coded in the DFTRMLST macro that defined the list: OPENLST, WRAPLST, DIALST, SSALST, SSAWLST, AUTOLST, AUTOWLST, or SWLST. (IDLST, BSCLST and WTTALST are invalid operands.)

listposition
: Specifies the original relative position in the list of the entry to be changed. Code a 1 if the first entry is to be changed, 2 if the second entry, etc.

numchars
: Specifies the number of polling or addressing characters in each entry of the list. This operand may be omitted if listype is SSALST, SSAWLST, or SWLST.

action
: Specifies the action to be performed on the entry:

- For listypes other than SWLST:

  SKIP
  indicates that polling or addressing is to be suspended.

  ACTIVATE
  indicates that polling or addressing is to be resumed.

- For a listype of SWLST:

  The following operands specify the action to be performed when the ID ENQ sequence is received from a remote station on a Read Connect operation.

  ACTIVATE
  specifies that BTAM is to send the ID ACK-0 sequence contained in the idsent field of the answering list, and then read a message block, if any. ACTIVATE sets the control byte to X'00'.

  DISC
  specifies that BTAM is to send the disconnect signal (DLE EOT) and then break the line connection. (The two commands that perform this function are part of the Read Connect channel program.) BTAM then restarts the

channel program at the Enable command to await a new call. DISC sets the control byte to X'01'.

POST
specifies that BTAM is to post the Read Connect operation complete. The user program then must take the appropriate action. POST sets the control byte to X'02'.

Return Codes: After you issue a CHGNTRY macro with a listype of SSALST, SSAWLST, AUTOLST, or AUTOWLST only, BTAM indicates the result of the operation, by means of a return code in register 15:

Code	Meaning
X'00'	The requested action was performed, or it was already performed (i.e.,polling or addressing was already suspended or resumed).
X'04'	The requested action was not performed, because the terminal list is in use by a Read or Write operation.
X'08'	The requested action was not performed, because the value coded in the listposition operand exceeded the number of entries in the list, i.e., no such entry exists.

This chapter describes how to construct buffer pools, obtain buffers through both programmer buffering and dynamic buffering, and release buffers after use. Dynamic buffering for Read and for Write operations is differentiated.

CONSTRUCTING BUFFER POOLS

If you intend to use buffers for holding input and output messages, a buffer pool must be constructed in one of several ways, as illustrated by Figures 7 through 10. Only those operands of concern in constructing buffer pools are shown. The BUILD, GETMAIN and GETPOOL macro instructions mentioned below are fully explained in the Supervisor and Data Management Macro Instructions publication.

Using the BUILD Macro Instruction

First, reserve a storage area at assembly time using DC or DS instructions, or issue a GETMAIN macro instruction to obtain the space. The area must begin on a fullword or doubleword boundary, and must contain enough space for an eight-byte buffer control block and the number of buffers needed.

Then issue a BUILD macro instruction specifying the number of buffers, their length, and the address of the area reserved at assembly time or obtained by the GETMAIN macro (GETMAIN provides the address, in a register, of the area it has obtained). The BUILD macro constructs the buffer control block and the buffer chain.

The length of each buffer must be four bytes longer than the length of the data to be placed in the buffer, because BTAM uses the first four bytes of each buffer as a link field containing the address of the next buffer. If this caution is not observed, the data, when placed in the buffers, may overlay the link field, which will destroy the link addresses and thus cause loss of data.

In the DCB macro instruction for each line group that is to use this buffer pool, specify the address of the buffer control block (BUFCB operand).

Figures 7 and 8 show examples that use DS statements and the GETMAIN macro.

Using the GETPOOL Macro Instruction

You may issue a GETPOOL macro instruction either before opening the data control block to be associated with this buffer pool or during the DCB exit routine. In the GETPOOL macro specify the address of the data control block and the number and length of the buffers you need. You must also specify the buffer length in the DCB macro (BUFL operand).

GETPOOL obtains sufficient storage to accommodate the pool, structures the buffer control block and the buffer chain, and places the buffer control block address in the data control block. See Figure 9.

BTAM Construction of Buffer Pools

If you wish for BTAM to provide the buffer pool automatically, you simply specify the number of buffers (BUFNO) and their length (BUFL) in the DCB macro for the line group that is to use the buffer pool. During the opening of the data control block, BTAM uses the operating system data management facilities to obtain main storage for the buffer pool, and then structures it. See Figure 10.

Once a buffer pool has been constructed, you can either request buffers yourself, before the Read or Write operation that will use them (programmer buffering), or let BTAM obtain them automatically (dynamic buffering).

PROGRAMMER BUFFERING

To obtain buffers yourself, issue a REQBUF macro instruction, specifying how many you need. Then check the return code in register 15 to determine whether all of the buffers you requested, some of them, or none of them are available to you. If any are available, REQBUF provides, in a register you have designated, the address of the first buffer. Simply specify this address in the READ or WRITE macro instruction. In the case of a WRITE macro, you move the message to be written into the buffers, beginning at the address of the first buffer.

In moving an output message into a buffer chain, remember that each buffer begins with a fullword link field. You must fill each buffer individually, inspecting the link field each time to learn the location of the next buffer. The address of the buffer, plus four bytes, yields the address where the message data should begin.

After you issue a REQBUF macro instruction, the return code in register 15 may indicate that only some of the buffers you requested are available, or none of them.

If some buffers are available, they are assigned to you. The address of the first one is in the register you designate, and register 0 indicates how many of the buffers that you requested were unavailable.

The action you take when the full number of buffers is not available depends on your application. Either use the number of buffers supplied (if any) and issue another REQBUF for the remainder; or, issue a RELBUF macro to release the ones supplied to you and reissue the REQBUF for the original number of buffers you requested. If the insufficient-buffer condition occurs infrequently, the cause is probably a momentary peak of activity on several lines at once. In this case, you will most likely obtain the buffers you need the next time you issue the REQBUF macro. On the other hand, frequent recurrence of this condition indicates that you should increase the number of buffers in the pool, as the amount of transmission activity on the lines using the pool exceeds the present capacity of the pool.

DYNAMIC BUFFERING

To be able to use dynamic buffering for a line group, you must specify BFTEK=D in the DCB macro instruction for the line group. Because channel programs differ for dynamic buffering and programmer buffering, and all lines in a line group use the same channel programs, you must use either dynamic buffering or programmer buffering for all lines in the group; you cannot use dynamic buffering for some lines, and programmer buffering for others.

Note: Dynamic buffering cannot be used for the local 3270 display system. If dynamic buffering is specified, the specification is ignored.

Read operations and Write operations employ dynamic buffering somewhat differently.

READ OPERATIONS

The first buffer for a Read operation may be obtained in one of two ways: either you supply the buffer yourself, by giving its address in the area operand of the READ macro, or you let BTAM provide the first buffer by coding 'S' as the area operand. BTAM places the address of the first buffer it obtains in the DECAREA field of the DECB for the line. This tells you where the received message begins. Regardless of which method you choose BTAM automatically obtains all subsequent buffers needed to contain the data being received. If you provide the first buffer yourself, BTAM automatically places the address of the first buffer it provides in the high-order fullword of your buffer and reads data into your buffer beginning at the second fullword.

An advantage of supplying the first buffer yourself is that it need not be a buffer from the buffer pool; it can be an area you have defined in your program as the place where all incoming messages begin; this affords you the convenience of always beginning your message processing at the same main storage address. Another advantage is that this area can be small compared to the size of your buffers allowing short messages to be read into this small area rather than into a regular buffer. Improved buffer utilization results, especially when the pool consists of a small number of large buffers.

After each buffer is full, it is posted complete. The first word of each buffer is treated as an event control block (ECB). A completion code is set in the high-order byte of the ECB, and the address of the next buffer is placed in the three low-order bytes.

The user program may wait for the entire message block to be read by issuing a WAIT macro for the primary ECB, in the same manner as is done without dynamic buffer allocation. Alternatively, the user program may wait for each buffer to be posted complete. This is accomplished by obtaining the address of the first buffer from the DECB and using that address as the ECB address in a WAIT macro instruction. After the first wait completes, the user program may obtain the address of the second buffer from the chain address field of the first buffer and issue a WAIT macro instruction for the second buffer. Succeeding buffers are waited for in a similar manner. After each buffer completes, the user program must check for a zero chain address, which indicates that it is the last buffer in the chain.

As the Read operation progresses, BTAM obtains buffers successively until it detects the receipt of an ending character such as ETB, ETX, or EOT. When this occurs, BTAM does not obtain any more buffers. If by the time the ending character is received BTAM has obtained another buffer, BTAM releases that buffer automatically, unless the ending character is in the last byte of the current buffer. In this event, you must release the extra buffer yourself. You may check for this condition in one of two ways.

1. Compare the residual count in the DECCOUNT field against the buffer length in the DCBBUFL field, minus four. If count and length-minus-four are equal, the last buffer BTAM obtained for the Read operation is unused. (This method cannot be used if the Read operation includes the Reset function, e.g., the Read Initial and Reset (TIR) option.)

2. Test the last byte of the next-to-last buffer for an appropriate ending character. If one is present, the last buffer is unused.

When you detect an unused buffer, release it with a RELBUF macro and place zeros in the low-order three bytes of the high-order word of the next-to-last buffer (i.e., the one containing the ending character), to indicate that this buffer is the last one in the chain. If you are waiting on buffers, do not release the unused buffer until it is posted.

In the channel programs for Read operations using dynamic buffering, each Read Text command is followed by a Read Skip command. When the Read Text command is executed, a program controlled interrupt (PCI) occurs. This causes BTAM to obtain another buffer, place its address in the next Read Text command, and change the Read Skip command to a transfer-in-channel (TIC) command pointing to the next Read Text command. When the first buffer is filled, incoming data begins filling the buffer just obtained. The same action occurs as each Read Text command is executed.

The action just described represents the normal case in which BTAM is able to obtain the next buffer in time to receive data from the line. Occasionally, however, BTAM may be unable to obtain the next buffer in time. Should this occur, the Read Skip command following the Read Text command remains unchanged. The Read Skip receives, but does not place in main storage, all data received from the line after the current buffer is full. In this way, the line is cleared of incoming data. The Read operation ends when BTAM detects an ending character, posts the operation as normally completed (X'7F') in the event control block (DECSDECB), and turns on bit 4 of DECFLAGS to indicate that part of the incoming message has been lost in the manner described. By checking this bit after each Read operation using dynamic buffering, you can detect the condition and take appropriate action; normally, you would release the buffers and send a negative response in reply to the message, causing the remote station to resend it.

As indicated under Programmer Buffering, frequent unavailability of buffers may be caused by a buffer pool that is too small to satisfy the demands made upon it. Increasing the number of buffers should solve the problem.

Caution: If the CPU is stopped while operations involving dynamic buffering are in progress, message data may be lost, as the program-controlled interrupts (PCI) required to obtain successive buffers are not handled when the CPU is stopped.

```
         ...
         BUILD     BFRPOOL,20,100            BUILD BUFFER POOL
         ...
         OPEN      (LINEGP1,,LINEGP2)        OPEN LINE GROUPS
         ...
---------------------------------------------------------------------
         ...
ENDJOB   CLOSE     (LINEGP1,,LINEGP2)        CLOSE LINE GROUPS
         ...
         RETURN
LINEGP1  DCB       BUFL=100,BUFCB=BFRPOOL,...
LINEGP2  DCB       BUFL=100,BUFCB=BFRPOOL,...
BFRPOOL  DS        D                         BUFFER CTL BLOCK SPACE
         DS        500F                      2000-BYTE BUFFER AREA
```

Figure 7. Constructing Buffer Pools Using DS and BUILD

```
         ...
         USING      IHADCB,DCBREG          ESTABLISH DCB  ADDRESSABILITY
         ...
         GETMAIN    R,LV=2008              OBTAIN STORAGE FOR POOL
         LR         POOLREG,1              OBTAIN ADDRESS OF POOL
         ...
         BUILD      (POOLREG),20,100       BUILD BUFFER POOL
         ...
         LA         DCBREG,LINEGP1         PLACE ADDRESS OF
         ST         POOLREG,DCBBUFCB       BUFFER POOL IN
         LA         DCBREG,LINEGP2         LINE GROUP
         ST         POOLREG,DCBBUFCB       DCB'S
         ...
         OPEN       (LINEGP1,,LINEGP2)     OPEN LINE GROUPS
         ...
-------------------------------------------------------------------------
         ...
ENDJOB   CLOSE      (LINEGP1,,LINEGP2)     CLOSE LINE GROUPS
         LR         1,POOLREG              PROVIDE ADDRESS OF POOL
         FREEMAIN   R,LV=2008,A=(1)        RELEASE STORAGE
         ...
         RETURN
LINEGP1  DCB        BUFNO=10,...
LINEGP2  DCB        BUFNO=10,...
         DCBD       DSORG=BX
```

Figure 8. Constructing Buffer Pools Using GETMAIN and BUILD

```
         ...
         GETPOOL    LINEGP1,10,100         BUILD BUFFER POOL
         ...
         GETPOOL    LINEGP2,8,120          BUILD BUFFER POOL
         ...
         OPEN       (LINEGP1,,LINEGP2)     OPEN LINE GROUPS
         ...
-------------------------------------------------------------------------
         ...
ENDJOB   CLOSE      (LINEGP1,,LINEGP2)     CLOSE LINE GROUPS
         ...
         FREEPOOL   LINEGP1                RELEASE BUFFER POOLS
         FREEPOOL   LINEGP2
         ...
         RETURN
LINEGP1  DCB        BUFL=100,...           SPECIFY BUFFER LENGTH
LINEGP2  DCB        BUFL=120,...
```

Figure 9. Constructing Buffer Pools Using GETPOOL

```
         ...
         OPEN       (LINEGP1,,LINEGP2)     OPEN LINE GROUP AND BUILD POOLS
         ...
-------------------------------------------------------------------------
         ...
ENDJOB   CLOSE      (LINEGP1,,LINEGP2)     CLOSE LINE GROUPS
         ...
         RETURN
LINEGP1  DCB        BUFL=100,BUFNO=10      SPECIFY BUFFER LENGTH
LINEGP2  DCB        BUFL=120,BUFNO=8
```

Figure 10. Constructing Buffer Pools Automatically

WRITE OPERATIONS

Whereas in Read operations the main storage locations of individual buffers are unknown to the programmer until BTAM links them into a chain, in Write operations the chain must already have been formed from buffers whose locations and contents are known to the programmer. You must, therefore, always specify in the Write operation the address of the first buffer in the chain whose contents are to be transmitted. As the Write operation progresses, BTAM provides to the operation the address of each of the remaining buffers in the chain.

Normally, you will have obtained the buffers for the Write operation by means of a REQBUF macro; or you will have obtained them dynamically during a preceding Read operation, when you wish to send the same data you received during the Read. A Write operation ends when BTAM detects an ending character, or when all the data in the last buffer has been transmitted, whichever occurs first. The length you specify in the WRITE macro must be great enough to encompass the number of characters in the last buffer, including the ending character or character sequence. If, instead of specifying the length, you code 'S' as the length operand, the ending character must be in the last buffer in the chain.

After each buffer is transmitted, it is posted complete, in the same manner as for Read operations. The user program may wait either for the entire message block to be transmitted or for each buffer, in the same manner as for Read operations. Once the message is successfully transmitted, the buffer chain can be returned to the pool with the RELBUF macro instruction.

BUFFER MANAGEMENT MACRO INSTRUCTIONS

REQBUF (Request Buffer) Macro Instruction

REQBUF is used to obtain one or more buffers from a buffer pool that has been constructed before or during opening of a line group data set.

When you are using programmer buffering, you may issue a REQBUF macro to obtain one or more buffers in which data can be received from a line (Read operations) or in which to build or move an output message (Write operations).

When you are using dynamic buffering, BTAM automatically obtains buffers for Read operations, so you do not issue a REQBUF macro to obtain them. For Write operations, however, use of REQBUF is the same as for programmer buffering.

The buffers provided are not necessarily in consecutive main storage locations. They are chained together, the link field of each containing the address of the next. The link field of the last buffer in the chain contains zeros.

Name	Operation	Operand
[symbol]	REQBUF	dcbaddr,returnreg, [count]

dcbaddr
Specifies the address of the data control block with which the buffer pool is associated.

returnreg
Specifies a general register (2 through 12) into which you wish BTAM to return the address of the first buffer to be provided.

count
Specifies the number of buffers you are requesting.
If you specify one of the registers 2 through 12, you must previously have loaded the count into the low-order byte of that register; the high-order bytes are ignored.
If you specify register 0, you must previously have loaded the count into the high-order byte of the register; the low-order bytes must contain zero.
If you omit this operand, BTAM provides one buffer, i.e., the link field contains zero.

Return Codes: After you issue a REQBUF macro, the low-order byte of register 15 contains a return code indicating the result of the buffer request. (The three high-order bytes of the register contain zero.) The return code, in hexadecimal notation, is one of the following:

00 Normal return. BTAM has provided the total number of buffers you requested. The return register contains the address of the first one.

04 Partial fulfillment of request. You requested more buffers than are currently available in the pool. All those available were provided. The return register contains the address of the first one.

08 No buffers available. The buffer pool had been exhausted at the moment of your request. The return register and register 0 contain zero.

0C No buffer pool. The request cannot be filled because no buffer pool is associated with the data control block you have specified in the macro.

10 No buffer routine. The request cannot be filled because the BTAM buffer management routine has not been included in your program. (The routine is automatically included if you have specified BFTEK=D in the DCB macro, or if the data control block contains the address of a buffer control block.)

Programming Note: If the buffer request was partially filled (return code is 04), the low-order byte of register 0 contains the count of the number of buffers not provided. (The three high-order bytes contain zeros.)

When the REQBUF macro instruction is used for the local 3270 display system, an entire message must fit into one buffer.

RELBUF (Release Buffer) Macro Instruction

RELBUF is used to return to the buffer pool one or more buffers obtained by a REQBUF macro or automatically during dynamic buffering. Failure to issue this macro instruction following Read and Write operations for which buffers have been obtained will ultimately result in exhaustion of the buffer pool. RELBUF releases each buffer in the chain, beginning with the one whose address you specify and ending with the one whose link field contains zero (i.e., the last buffer).

Name	Operation	Operand
[symbol]	RELBUF	dcbaddr,bufferaddr

dcbaddr
    Specifies the address of the data control block associated with the buffer pool to which the buffers are to be released.

bufferaddr
    Specifies a general register (2 through 12) into which you must previously have placed the address of the first buffer to be released.

Return Codes: After you issue a RELBUF macro, the low-order byte of register 15 contains a return code indicating the result of the operation. (The three high-order bytes of the register contain zeros.) The return code, in hexadecimal notation is one of the following:

00 Normal return: The specified buffers have been returned to the pool.

04 Already returned: The first buffer of the chain to be released has already been returned to the pool (or has never been obtained from the pool).

0C No buffer pool: The buffer release cannot be accomplished because no buffer pool is associated with the data control blocks you have specified in the macro.

10 No buffer routine: The buffer release cannot be effected because the BTAM buffer management routine has not been included in your program.

Programming Note: If you wish to release a different number of buffers than you obtained by a REQBUF macro or by dynamic buffering (assuming the first buffer to be released is the same as the first buffer that was obtained), you will have to place zeros in the link field of the last buffer you wish returned. Be sure to retain the address of the buffer that follows the last one you return, as it will become the first of the remaining buffers in the original chain.

As pointed out in the first chapter, in the discussion of how information is represented in various parts of a teleprocessing system, it is the programmer's responsibility to perform code conversion between transmission code and the internal code of the central computer, if the application requires it.

BTAM provides a translation routine and a set of translation tables that convert between EBCDIC and the transmission code or codes employed by the types of remote stations supported by BTAM. Some terminal types can be furnished with any of several character sets; BTAM provides translation tables for the more common sets. (In most cases the sets vary by only a few characters.) When a remote station in your configuration uses a character set not directly supported by a BTAM-provided translation table, you can easily modify an existing table to accommodate that station. Alternatively, you can define an entirely new table (but do not give it the same name as a BTAM-provided table). You must format any table you define according to the requirements of the System/360 Translate (TR) instruction (see the Principles of Operation manual).

If you wish to refer to a BTAM-provided translation table after assembling it into your program (for example, to modify the table via a MVC instruction or to use it in conjunction with the TR instruction), you must refer to the table by the name IECTxxxx, where the x's represent the four-character table name as shown in Table 4. In referring to the table with the TRNSLATE macro, however, you need specify only the four-character table name. Table 4 lists the translation tables provided by BTAM.

At the end of this publication are two sets of code tables. Appendix H is a code correspondence chart that shows for each of the 256 EBCDIC bit patterns the corresponding character (and its transmission code bit pattern) to or from which the BTAM-provided translation tables convert the EBCDIC character. Full understanding of this chart requires that you read the explanatory maerial preceding it.

Appendix I shows for each of the 256 possible bit patterns in a System/360 byte the character represented by that pattern in each of the transmission codes and in EBCDIC. This chart is useful in interpreting the contents of main storage locations.

ASMTRTAB (Assemble Translation Table) Macro Instruction

ASMTRTAB assembles into a program one or more BTAM-provided translation tables. You may code all table names in one ASMTRTAB, and you need code only one ASMTRTAB regardless of the number of lines and line groups for which the table is needed. Code the macro among the program constants, not in the middle of executable code.

Note: The ASMTRTAB macro instruction is not used for the local 3270 display system.

Name	Operation	Operand
(Omit)	ASMTRTAB	tablename,...

tablename
: Specifies the BTAM-provided translation table or the table you wish to assemble into your program. Code any table name listed in Table 4. Table names may be coded in any sequence.

Example: If you wish to perform code translation between EBCDIC and 1030 code, and between EBCDIC and TRANSCODE, code:

```
          ASMTRTAB  RC30,SD30,RC80,SD80
```

TRNSLATE Macro Instruction

TRNSLATE translates data in main storage from transmission code to EBCDIC (for received data) or from EBCDIC to transmission code (for data to be transmitted). Code TRNSLATE at each point in your program where translation is required.

Note: The TRNSLATE macro instruction is not used to translate between transmission code and EBCDIC for the local 3270 display system.

Name	Operation	Operand
[symbol]	TRNSLATE	[dcbaddr],tablename, area,length

dcbaddr
Specifies the address of the data control block for the line group. This operand is required if you code the length operand as 'S'; otherwise, it may be omitted.

tablename
Specifies the four-character name (e.g., RC50) of the BTAM-provided translation table to be used, or the name of your own translation table. (You must have previously assembled the indicated table into your program.)

area
Specifies the address of the main storage area in which the data to be translated is located. If dynamic buffering is used for the line group involved, the address specified by area must be a fullword boundary. If you use TRNSLATE to translate a chain of buffers you have defined, those buffers must be formatted and chained just like BTAM-provided buffers: the first fullword of each buffer contains the address of the next buffer (except that the first fullword of the last buffer contains zeros). Each buffer must begin on a fullword boundary.

length
Specifies the number of bytes to be translated, from 1 to 32,767. If you wish to translate the contents of a chain of buffers, code 'S' as the length operand. This causes the translate routine to use the buffer length given in the data control block.

Type of Remote Station	Transmission Code	Table Name
For incoming messages: (Translation from transmission code to EBCDIC):		
IBM 1030	EBCD/PTTC	RC30
IBM 1050	EBCD/PTTC	RC50
IBM 1060	BCD/PTTC	RF50*
IBM 2260	USASCII	RSCI
IBM 2740, 2741	BCD/PTTC	RB40, RU40*
	EBCD/PTTC	RC40, RF40*
	Correspondence code	RC41, RF41*
IBM S/360 (incl. Model 20)	USASCII	RASA
IBM System/3	USASCII	RASA
IBM 2770	USASCII	RASA
IBM 2780	USASCII	RASA
	6-bit Transcode	RC80
Remote IBM 3270	USASCII	RASA
AT&T 83B3, WU 115A	Baudot code	RCT1
WU TWX (Models 33,35)	TWX Code (even-parity)	RCT2
World Trade Telegraph terminals	ZSC3 code	RCT3
	ITA2 code	RCTW

Table 4. Code Translation Tables Provided by BTAM (Part 1 of 2)

Type of Remote Station	Transmission Code	Table Name
For outgoing messages (translation from EBCDIC to transmission code):		
IBM 1030	EBCD/PTTC	SD30
IBM 1050	EBCD/PTTC	SD50
IBM 1060	BCD/PTTC	SD60
IBM 2260	USASCII	SSCI
IBM 2740, 2741	BCD/PTTC EBCD/PTTC Correspondence code	SB40 SD40 SD41
IBM S/360 (incl. Model 20)	USASCII	SASA
IBM System/3	USASCII	SASA
IBM 2770	USASCII	SASA
IBM 2780	USASCII 6-bit Transcode	SASA SD80
Remote IBM 3270	USASCII	SASA
AT&T 83B3, WU 115A	Baudot code	SCT1
WU TWX (Models 33,35)	TWX code	SCT2
World Trade Telegraph Terminals	ZSC3 ITA2	SCT3 SCTW

1. Translation tables marked * convert both uppercase and lowercase alphabetic characters to uppercase EBCDIC equivalents (e.g., both A and a are converted to A); tables not so marked convert uppercase to uppercase and lowercase to lowercase (e.g., A to A and a to a).

2. Transmission code abbreviations used above:
   BCD = binary coded decimal
   EBCD = extended binary coded decimal
   PTTC = perforated tape and transmission code
   USASCII = USA Standard Code for Information Interchange
   ZSC3 = Figure Protected Code
   ITA2 = International Telegraph Alphabet No. 2

3. See General Note in Appendix H for discussion of TWX code parity.

Table 4. Code Translation Tables Provided by BTAM (Part 2 of 2)

The operations performed by a user's teleprocessing program preparatory to data transmission is called activating the system. Similarly, deactivating the system refers to the operations performed after all transmission has ceased. These operations largely consist of opening (activating) and closing (deactivating) the communications line group data sets.

PROGRAM INITIALIZATION

Before activating the TP system you must first perform the usual initialization steps required of any program that runs under the System/360 Operating System. These are as follows:

1. Using a SAVE macro instruction (or a Store Multiple instruction), store the contents of the general registers you will use in your program in a register save area, the address of which is in register 13 upon entry to your program.

2. Store the contents of register 13 in the second fullword of a save area you have defined in your program.

3. Load the address of your program's save area into register 13. (Save areas are required by most system macro instructions.) Unless you require register 13 for other purposes, you need to load it only at the beginning of your program.

See Figure 12 for an example of the foregoing linkage. More detailed information on the use and format of register save areas and on linkage conventions is contained in the OS Supervisor Services Guide, GC28-6646 and the OS Data Management Services Guide, GC26-3746.

In addition to these initialization steps, you should create dummy control sections (DSECTs) for the data control blocks and data event control blocks in your program, to allow you to refer symbolically to fields in these control blocks. To create these DSECTs, use the DCBD and IECTDECB macro instructions as shown in Figure 11. If the TP system includes BSC stations, a second operand, DEVD=BS, must appear in the DCBD macro; similarly, if the system includes World Trade telegraph terminals, code the second operand as DEVD=WT. If the system includes both kinds of stations, code DEVD=(BS,WT).

Code the DCBD and IECTDECB macros at the end of the control section (CSECT) in which they appear.

```
YOURPROG CSECT
            .
            .
            .
         USING   IHADCB,DCBREG
         USING   IECTDECB,DECBREG
            .
            .
            .
         DCBD    DSORG=BX
         IECTDECB
```

Figure 11. Establishing Addressability for DCBs and DECBs

OPENING AND CLOSING LINE GROUP DATA SETS

Before you can perform data transmission operations over a line, you must open, or activate, the line group data set encompassing that line, by means of an OPEN macro instruction. When you issue an OPEN macro, an OPEN routine establishes and initializes various internal control blocks, and loads from the system library those routines and tables needed for BTAM to construct the channel programs required by subsequent READ and WRITE macros. The Open routine also "conditions" the communications line adapters within each transmission control unit (TCU) associated with the line group. Conditioning a line adapter makes the line attached to it ready for data transmission.

The fact that you have issued an OPEN macro does not guarantee that the line group is open. The DCB for the line group has a bit, called the Open flag, that you can check to determine whether the line group is open. The Open flag is bit 3 of the DCBOFLGS field; if it equals 1, the line group is open.

If after you issue the OPEN macro, the Open flag is still 0, there is probably a coding error; most likely, the DD (data definition) card for the line group contains the wrong line group name.

```
YOURPROG CSECT
         SAVE      (14,12)              1.  SAVE REGISTERS IN CALLING
*                                           PROGRAM'S SAVE AREA
         LR        BASEREG,15
         USING     YOURPROG,BASEREG
         ST        SAVEREG,SAVEAREA+4   2.  SAVE REG 13 IN 2ND FULLWORD
         LA        SAVEREG,SAVEAREA     3.  LOAD YOURPROG SAVEAREA
*                                           ADDRESS
BASEREG  EQU       12
SAVEREG  EQU       13
         .
         .
         .
BEGIN    EQU       *
         .
         .
         .
SAVEAREA DS        18F
```

Figure 12. Initializing Your Program

Even if the Open flag is 1, however, one or more lines in the line group can be unready for transmission because the line adapter was not successfully conditioned. If this occurs because the TCU power is off or if the TCU is off-line, the operating system prints, on the console, error message IEC804A, and enters Wait state. This message identifies the condition and requests a response from the console operator.

He replies CONT (Continue) if he wishes the Open function to be retried, POST if he does not. The usual procedure is to correct the abnormal TCU condition, then reply CONT. Or, he may ignore the condition and reply POST, so that the user program can proceed with operations on unaffected lines.

The console message is issued only if the line adapter to be conditioned was caused by a TCU power-off or off-line condition. If unsuccessful conditioning occurs for some other reason (e.g., TCU malfunction), the fact that the line has not been opened becomes evident when the first READ or WRITE macro issued for that line results in a return code of X'14'. For this reason, the user program should check for this return code after the first READ or WRITE macro following opening of the line.

BTAM provides the LOPEN (Line Open) macro instruction for use in opening (i.e., conditioning the line adapter for) a single line in a line group. LOPEN is intended for use following a return code of X'14'.

Depending on your application, you may wish to open all line groups at once, or to open different groups at successive intervals during the day. Opening line groups at different times would be appropriate, for instance, when the remote stations connected to one group are located in a different time zone from those connected to another group.

After completion of data transmission over all lines in a line group, you may close the line group by means of a CLOSE macro instruction. If BTAM provided a buffer pool during opening of the line group (see the Buffer Management chapter), you must issue the CLOSE macro only after you have no further use for the contents of any of the buffers in that pool. This is necessary because when you close the line group, BTAM relinquishes the main storage area occupied by the pool, and various pointers to buffers no longer exist.

OPEN Macro Instruction

OPEN completes the initialization of the data control block representing the line group data set, builds a buffer pool, if you specify in the DCB macro that this be done, and loads from the system library those routines and tables necessary for BTAM to construct the appropriate channel programs. As explained earlier, the Open routine also conditions each transmission control unit line adapter connected to a line in the group.

See Figure 13 for the format of the OPEN macro instruction.

A single OPEN macro can activate any number of line groups and any other data sets defined in your program, including those for other access methods.

Example: To open two line group data sets and three BSAM data sets (one on magnetic tape, two on direct access devices), you could code a single OPEN macro as follows:

```
OPEN1    OPEN     (LG1050,,LG2740,,
                  TAPELOG,(OUTPUT),
                  MSGFILE1,(INOUT,LEAVE),
                  MSGFILE2,(OUTPUT))
```

LG1050 and LG2740 are the two line group data sets; the second comma following each of these operands indicates the absence of volume-positioning option parameters, which are not appropriate for communications line groups. The remaining operands are representative of data set addresses and volume-positioning options for the three BSAM data sets. (See the Supervisor and Data Management Macro Instructions publication for information on coding OPEN macros for non-line group data sets.)

No return code is provided following an OPEN macro instruction; as explained earlier, you should check the Open flag in the DCB to see if the line group was successfully opened.

LOPEN Macro Instruction

LOPEN causes BTAM to issue commands that condition the transmission control unit line adapter for a specific line, when conditioning of the adapter was not successful during opening of the line group. It is appropriate to issue LOPEN after receiving a return code X'14' following issuance of a READ or WRITE macro for the line. LOPEN causes the appropriate command (Set Address, Set Mode, or Enable) to be sent to the line adapter.

(LOPEN may also be used to reestablish data set synchronism for a line using an IBM 3977 Model 2 modem (data set), as follows. When the modem loses synchronism, transmission errors (i.e., a NAK response from the remote station or a time-out error) will occur during Write operations. When errors of these kinds occur, it is appropriate to issue an LOPEN macro, which, by disabling the line and then enabling it or setting the mode, causes the modem to regain synchronism.)

Name	Operation	Operand
[symbol]	LOPEN	decbaddr

decbaddr
Specifies the address of the data event control block associated with the line.

Programming Note: You should not issue a LOPEN macro from within a timer exit, since LOPEN uses the STIMER macro.

Return Codes: Upon return of control to your program, the low-order byte of register 15 contains a return code. Normal completion is indicated by X'00'. Abnormal completion is indicated by the following codes (hexadecimal):

04 The line was not successfully opened.

08 The specified line is busy.

0C The relative line number specified in the data event control block is larger than the number of lines in the line group.

10 The DCB for the line group is not open.

14 The request was rejected, because OLTEP was using the local 3270 device.

Only the first and last of these abnormal return codes, 04, and 14 will be encountered in a debugged user program; the other three result from program errors. If a code of 04 is returned after you issue an LOPEN macro, you may wish to notify the console operator that he should check the condition of the affected transmission control unit.

CLOSE Macro Instruction

CLOSE terminates the availability of a line group data set; frees the main storage space occupied by the buffer pool, if the pool was constructed by the Open routine; and frees the main storage space obtained by the Open routine for control blocks. CLOSE also causes the fields in the data control blocks to be restored to the condition they were in before the DCB was opened. Just as OPEN causes the TCU line adapters associated with the line group to be conditioned for use, CLOSE cancels the conditioning. For this reason, if you issue a CLOSE macro instruction while data transfer is still in progress over one or more lines in the line group, unpredictable loss of data can result. You should therefore close the line group only after all message traffic has ceased. See Figure 13 for the format of the CLOSE macro.

A single CLOSE macro can deactivate any number of line groups and any other data sets defined in your program (including those for other access methods), in the same way an OPEN macro can activate them.

Name	Operation	Operand
[symbol]	{OPEN / CLOSE}	({dcb,,}...),[MF=L / MF=(E,listname)]

symbol

Specifies:

- For standard or execute macro format, the name of the first instruction generated by the macro. For these formats the use of symbol is optional.
- For list format, the name of the parameter list created by the macro. For this format, you must specify a name.

dcb

Specifies the name of the line group data set you wish to open or close.

MF=L (List format)

Specifies that a parameter list is to be created, containing the names of the data control blocks to be opened or closed. The function is not performed until you issue an OPEN or CLOSE macro of the execute format specifying the name of the parameter list.

MF=(E,listname) (Execute format)

Specifies that the open or close function is to be executed for the data sets contained in the parameter list specified by listname. You must previously have created the list with an OPEN or CLOSE macro of the list format (MF=L). If you wish to override certain parameters in the list, specify replacement parameters in the macro having the execute format. Code the replacement parameters in the positions corresponding to the locations of the parameters to be overridden.

(Standard format - MF operand omitted)

Specifies that both (1) a parameter list is to be created, containing the names of the data control blocks to be opened or closed, and (2) the open or close function is to be executed for the data sets contained in the created parameter list.

<u>Example:</u>

```
OPENLIST OPEN      (LG1050,,LG2740,,LG1130),MF=L
         .
         .
         .
         OPEN      (,,LG2260),MF=(E,OPENLIST)
```

The first macro creates a list; the second executes the Open function for data sets LG1050, LG2260, and LG1130.

Once you have defined a parameter list by either an OPEN or a CLOSE macro of the list or standard format, you may subsequently specify that list by both OPEN and CLOSE macros of the execute format.

Figure 13. Formats of OPEN and CLOSE Macro Instructions

LINE CONTROL

Communication between the central computer and remote stations requires a discipline called line control, as mentioned earlier in this publication. Given here is a summary of the control scheme used for various line configurations and types of remote stations.

Line control does not apply to the local 3270 display system, which uses attention interruptions to regulate communications between the central computer and local display stations. For more information, see "Attention Interruptions and Read Initial Operations" in the section "IBM 3270 Display System - Programming Considerations."

Contention System

In the most elementary form of line control each of the two stations at the ends of a point-to-point communications line gains use of the line by sending to the other station a special control character signifying the station's intention to begin transmission. The first station to initiate contact in this manner "seizes" the line and prevents its use by the other station until the first station has concluded its message transmission. If both stations should simultaneously try to initiate transmission, they are said to be contending for use of the line, hence the name contention system. In this kind of system some method is required for resolving a contention situation.

The action of requesting use of the line is sometimes called bidding for the line.

Centrally-Controlled System

In this kind of system, the central computer acts as a control station. That is, it initiates all contacts between all stations on a multistation (multipoint) line. It does this by periodically sending on the line a series of station identifiers, called polling characters or polling sequences. Each station on the line has a different polling sequence. Thus, although all stations receive all polling sequences, each station responds only to its own. This response indicates to the control station (the central computer) whether or not that remote station is ready at that moment to send a message. It sends a positive response if it is ready, a negative response if it is not. For some types of stations the polling sequence identifies a specific component of the station, as well as the station itself. In this case, the response indicates whether or not that particular component is ready to send a message.

Similarly, when the control station wishes to send a message to a remote station, it transmits an identifier sequence on the line. This is called addressing, or selection. Again, all stations receive the addressing characters, but only one responds. The addressed station returns to the computer a positive response if the station (and perhaps a specific component) is ready to receive a message.

In a system of this kind, the stations can be in one of two modes: control mode and text mode. The stations are all in control mode before a transmission begins, and in this mode they monitor the line for polling and addressing sequences. When a polled or addressed station responds positively, message transmission between the central computer and the remote station can begin. At this point, it is necessary to place all stations in text mode, so that any characters received by any station except the polled or addressed station are ignored. (If the other stations remained in control mode, any sequence of message characters that happened to constitute a polling or addressing sequence for one of the stations would activate that station.) Accordingly, each message begins with a special control character whose purpose is to cause the stations to enter text mode. Two characters used for this purpose are EOA (end of address) and STX (start of text). The type of station on the line determines which character is used.

At the end of a transmission, all stations on the line must be returned to control mode so that they can again respond to polling and addressing sequences. Another character or character sequence, called end-of-transmission (EOT), performs this function.

The function of returning the stations to control mode is often called resetting the line.

Switched Systems

In a switched system, contact must be established by one or the other of two stations: the central computer or the remote station. In some switched configurations either the computer or a remote station can call the other station; in others, only one or the other of these can make the call. User requirements determine which case applies.

When the computer initiates contact with the remote station, it performs the calling function, when it answers a call from a remote station, it performs the answering function.

Although a remote station can call the central computer at any time, the computer, to fulfill its function as control station, must be able to accept or reject a call. If it wishes to accept calls, it "enables the line", that is, conditions the transmission control unit to respond to calls over the given switched line termination. The user program determines which lines are to be enabled at any given moment. Conversely, to return the TCU to the state in which it will not respond to (i.e., answer) calls is called disabling the line.

If a remote station calls in on a line that is not currently enabled, or that is enabled but is occupied with another remote station, the calling station receives a busy signal, and contact is not established. The station must try again later.

Once the line connection is established, one of the preceding line control schemes --contention or centrally controlled -- takes effect just as on a nonswitched line. The scheme used is the same as that used for a nonswitched line for the particular type of stations involved.

ERROR DETECTION AND MESSAGE BLOCKING

Line control may also involve detection of transmission errors. For the types of remote stations for which this is possible, a character called end-of-transmission-block (ETB) (also called EOB, end-of-block) is sent following a sequence of text characters; this sequence is then called a message block. Whenever the sending station senses an ETB in the data it is sending, it follows that ETB with a check accumulation (VRC, LRC, or cyclic) and awaits a response from the receiving station. The receiving station compares the check character with the check character it has accumulated. If they match, indicating that it received the text without error, it sends a positive response (or acknowledgment) to the sending station. If they do not match, indicating a transmission error has occurred, it sends a negative response (acknowledgment) to the sending station. A positive response indicates that the sending station may continue with the next message block; a negative response tells it to resend the erroneous block.

CHANNEL PROGRAMS

The various line control functions are achieved by the central computer through a combination of equipment and programming. Generally, each discrete function, such as enabling or disabling the line and reading and writing message text and responses is effected by separate channel commands that, when combined in appropriate sequences in a channel program, perform the overall line control actions needed to establish contact, transmit messages and check for errors. Channel programs are generated by BTAM as directed by the READ and WRITE macro instructions issued in the user program.

MESSAGE TRANSMISSION

All message transmission is effected by Read and Write operations of various kinds, which in turn are produced by coding equivalently named macros in the user program. (For information about Read and Write operations for the local 3270 display system, see the section "IBM 3270 Display System - Programming Considerations.")

A Read or Write Initial operation establishes contact with the remote station and receives or sends the first message block. In establishing contact, the operation performs whatever functions are appropriate. That is, for a nonswitched line in a contention system, Read Initial first sends the character that signifies to the receiving station that the line is being seized by the sending station. In a centrally controlled system, the first function is to send a character or sequence that places all stations in control mode, as explained earlier. For a switched line, Read Initial either enables the line, if the operation is to continue when a remote station calls in, or it dials the remote station.

Following execution of whichever of the foregoing functions is appropriate, polling may take place, if required by the type of station involved. Then the first block of the message is read or written.

Once a Read or Write Initial operation has concluded, you generally issue as many READ or WRITE Continue macro instructions as necessary to receive or send the remaining blocks of the message.

If a Read operation receives an erroneous message block, you may undertake Read Repeat operation; the negative response sent by Read Repeat signifies to the remote terminal operator or to the remote computer program that he, or it, should resend the block in error.

Sometimes it is desirable to reverse the direction of message transmission during one transaction, or to exchange the roles of the receiving and sending stations. Read and Write Conversational operations permit this.

In binary synchronous communications it is sometimes desirable to send data in transparent mode. This means that any transmission code bit pattern can be sent as data, whereas in normal transmission certain patterns are recognized and responded to as line control characters. Read and Write transparent operations are available for this purpose.

These various operations can be combined in several ways. Inspection of the Read and Write operations for a specific type of remote station and line configuration will illustrate some of these ways.

Although in coding a user program, it is not usually necessary to understand all the details of the various commands that make up a channel program, each command is explained fully in the SRL publications pertaining to transmission control units. These publications are listed at the front of this manual.

User Program Analysis

Upon completion of each Read or Write operation, the user program must analyze the results of the operation to determine which Read or Write operation to perform next. Where the operation was successful and either message text or some expected response was received, the decision about the next operation depends largely on the kind of application. Sometimes, it may depend on the content of the received text. For example, in an application that involves transmission of fairly long messages, it is common practice to break the message into sequences of message blocks. It is then appropriate to send or receive the first block using a WRITE or READ Initial macro instruction, or one of the variants, such as WRITE Initial Transparent, for BSC, and then send or receive the rest of the blocks with WRITE or READ Continue macros, or variants.

An operation may end successfully, but with some exceptional condition. For example, a sequence of Read operations will end when a remote station sends an EOT after having sent a number of blocks of text. Since the user program probably does not know when to expect the last block of text, if message lengths vary, it should check after every Read operation for receipt of an EOT, which is considered an exceptional condition.

Some operations will end unsuccessfully, with an error condition of some kind, such as a parity error (data checks) in text or an invalid response. Again, the user program must analyze the results of each Read or Write operation to see if an error condition has occurred.

BTAM provides error recovery procedures (ERP) for automatically attempting to recover from errors. These are optional for start-stop lines, mandatory for BSC lines. It is only after BTAM ERP has attempted recovery and failed that the error condition is indicated to the user program. If ERP is successful in clearing the condition, BTAM posts the operation complete-without-error, and the user program is unaware that the error has occurred.

The chapter, Error Recovery Procedures and Error Recording, discusses the BTAM ERP facilities and suggested user analysis procedures.

Use of Line Control Characters

To achieve successful communication with any given type of remote station requires that the data stream between the central computer and the remote station contain the appropriate line control (also called data link control) characters and character sequences. A BTAM programmer must be concerned with the proper use of these characters. In message data received from a remote station, you may need to scan the input areas to determine the locations of control characters, and perhaps to remove them. In message data to be sent to a remote station, however, you must assure yourself that these characters are sent at the appropriate point in the transmission. Some control characters are sent automatically by BTAM, in a separate command within a channel program. Others you must place in the message output area. For example, when using a transparent-type Write operation to send data in transparent mode (i.e., to prevent the control units at the central computer and remote station from reacting to bit patterns that correspond to line control characters), you must place the DLE STX character sequence in the output area at the point where transparent transmission is to begin. You do not, however, place the ending sequence, DLE ETX (or DLE ETB), in the output area because, as inspection of the channel program shows, the command following the Write Text command sends these characters.

It is most important to be familiar with the usage of line control characters for the type of remote station for which you are coding Read and Write operations. The line control characters and their proper usage are defined in the Systems Reference Library publications pertaining to the various types of stations, and, in the case of binary synchronous communications, in the General Information publication for BSC. (These publications are listed in the Preface of this publication.) Line control character usage may vary depending on particular features or combination of features with which the stations are equipped.

The next two chapters of this publication contain descriptions of the READ and WRITE macro options available for each of the types of remote stations with which the central computer can communicate under BTAM control. In Start-Stop Read and Write Operations, these descriptions are arranged by type of station. In BSC Read and Write Operations, they are arranged by type of line configuration. This is done because the channel program for each type of operation is the same for any type of station (for a given line configuration).

The Reset Function: For many of the Read and Write operations listed there is an optional reset function. This simply means that if the Read or Write operation has progressed satisfactorily up to that point (i.e., message text was received or sent without error), one or two additional commands are executed that reset the station to control mode, and, for switched line operations, that break the line connection. This is the only difference between a reset and a non-reset operation, and for this reason is not stated explicitly in each description. The reset function is not performed if a permanent error occurred during the operation.

Terminal Lists

The description of the DFTRMLST macro instruction earlier in this publication explains all of the operands of that macro. In the next two chapters, each section covering a type of remote station or a line configuration indicates which type of terminal list you must define for Read and Write operations and shows what operands to code in the DFTRMLST operand field to obtain that list. See the explanation of the DFTRMLST macro for the meanings of the operands, and see Appendix A for format illustrations and examples.

Data Event Control Block

The parameters BTAM needs to perform a Read or Write operation are contained in a data event control block (DECB). Some of these parameters are:

- The type of Read or Write operation (e.g., Initial, Continue, Conversational);
- The address of the data control block (DCB) for the line group encompassing the line over which the operation is to take place;
- The relative line number of the line involved;
- The address of the terminal list entry containing the information necessary to establish contact with the remote station; and
- The addresses of the input or output areas to contain the message text.

The DECB also contains fields in which the results of the Read or Write operation are indicated. Among these fields are:

- An event control block (ECB), in which a standard completion code is placed upon conclusion of the Read or Write operation;
- A response field (DECRESPN), into which responses from the remote station to polling and addressing are received; and
- Fields containing specific indicators of the results of the operations: DECSENS0 (sense information); DECFLAGS (condition flags); DECERRST (error status) and DECCSWST (channel status word status byte).

The format of the DECB and the contents of its fields are given in Appendix B.

One DECB is required for each communications line; more than one can be provided, if desired.

DECBs are created by READ and WRITE macro instructions as follows. A macro of the list form (specified by the keyword operand MF=L) reserves space for a DECB and fills in certain of its fields with the parameters provided by the macro. This is done at assembly time, and is the sole function of the list form macro. That is, the macro does not perform a Read or Write operation. If you define a DECB in this way, you must code the macro among the program constants (or create your own linkage around it), since a macro of the list form does not generate executable code.

In order to perform a Read or Write operation using a DECB created by the list form of the macro, you issue a READ or WRITE macro of the execute form, specified by the MF=E keyword operand. This form of macro does not establish a DECB; it executes the Read or Write operation using an existing list. In this macro you may specify which, if any, of the parameters in the original DECB you wish to change. For example, if you wish to issue a series of WRITE macro instructions, all of which require the same DECB parameters except for the entry parameter, it would be appropriate to issue one WRITE (or READ) macro of the list form to establish the DECB. Then you would code the other WRITE macros in the execute form, and in each one specify only the entry operand, of those operands that are optional.

An alternate method is to code the standard form of the READ or WRITE macro instruction, by omitting the MF keyword operand. A macro of this type generates both a DECB and the executable code required to perform the Read or Write operation.

Just as you issue a macro of the execute form referring to a DECB defined by a macro of the list form, you may issue an execute-form macro that refers to a DECB generated by a previous macro of the standard form.

An important point to remember in using the same DECB for a sequence of Read or Write operations is that the contents of many of the fields will change with each issuance of a macro or execution of a Read or Write operation. This means that at the conclusion of each Read or Write operation you should do whatever checking of DECB fields is necessary before you issue the next macro that will refer to the same DECB.

Not all of the READ and WRITE macro operands are optional. Regardless of the macro form, you must provide the address of the DECB and the operation type. In the standard form you must always code the DCB address and the relative line number.

READ and WRITE Macro Instructions

READ and WRITE macro instructions produce the Read and Write operations that achieve message transmission. You issue one of these macro instructions each time you wish to receive a message from a remote station, send a message to a remote station, or perform any of several other functions related to message transmission, such as sending and receiving responses, disabling or disconnecting a switched line, etc.

In the READ or WRITE macro you specify:

- The line group and specific line within that group over which the operation is to occur.
- The address of a terminal list, or an entry in that list, that contains the information BTAM needs to establish contact with a station. Examples of this kind of information are telephone numbers, polling and addressing sequences, and identification sequences.
- The type of Read or Write operation to be performed (Read Initial, Write Continue, etc.)
- The address of the data event control block (DECB) that the READ or WRITE macro is to define, or the address of an existing DECB that the operation will use.
- The addresses of input and output areas into which or out of which message text is to be received or sent.

Each of these parameters is discussed in the explanation of the operands.

Table 5. READ and WRITE Options for Start-Stop (Part 1 of 2)

OPTION	TYPE CODE	1030	1050 (nonsw)	1050 (switched)	1060	2260	83B3 115A	TWX 33/35	WT Teleg.
READ Initial	TI	X Note	X Note	X	X Note	X	X	X	X
READ Initial with Reset	TIR	X Note	X Note	X	X Note	X		X	
READ Continue	TT	X Note	X	X	X Note	X			X
READ Continue with Reset	TTR	X Note	X	X	X Note	X			
READ Continue with Leading Acknowledgment	TTA								
READ Continue with ID Exchange	TE								X
READ Conversational	TV			X			X		
READ Conversational with Reset	TVR			X			X		
READ Repeat	TP	X Note	X	X	X Note	X			
READ Repeat with Reset	TPR	X Note	X	X	X Note	X			
READ Buffer	TB					X			
READ Buffer with Reset	TBR					X			
READ Skip	TS	X	X	X	X	X	X	X	
WRITE Initial	TI	X	X	X	X	X	X	X	X
WRITE Initial with Reset	TIR	X	X	X	X	X	X	X	
WRITE Initial Optical	TIO								
WRITE Invitational Optical	TCO								
WRITE Continue	TT	X	X	X		X			X
WRITE Continue with Reset	TTR	X	X	X		X			
WRITE Continue Conversational	TTV		X*	X					
WRITE Conversational	TV			X			X		
WRITE Conversational with Reset	TVR			X			X		
WRITE Conversational Optical	TVO								
WRITE at Line Address	TL			X		X			
WRITE at Line Address with Reset	TLR					X			
WRITE Erase	TS					X			
WRITE Erase and Reset	TSR					X			
WRITE Break	TB						X		
WRITE Positive Acknowledgment	TA	X	X	X	X	X			
WRITE Negative Acknowledgment	TN	X	X	X	X	X		X	
WRITE Disconnect	TN								

Note: Options for which Auto Poll channel programs are generated if the IODEVICE system generation macro instruction for the time specified FEATURE=AUTOPOLL.

*Write TTV cannot be used if Auto Poll is specified (i.e., FEATURE=AUTOPOLL in IODEVICE macro for the line).

Table 5. READ and Write Options for Start-Stop (Part 1 of 2)

OPTION	TYPE CODE	2740	2740C*	2740D*	2740 DC*	2740 DT*	2740 DTC*	2740 S*	2740 SC*	2740 CO*	2740 DCO*	2741 NS*	2741 SW*
READ Initial	TI	X	X	X	X	X	X	X Note	X Note	X	X	X	X
READ Initial with Reset	TIR		X	X	X	X	X		X Note	X	X		
READ Continue	TT		X		X		X		X	X	X		X
READ Continue with Reset	TTR		X		X		X		X	X	X		
READ Continue with Leading Acknowledgment	TTA									X	X		
READ Continue with ID Exchange	TE												
READ Conversational	TV			X	X	X	X				X		X
READ Conversational with Reset	TVR			X	X	X	X				X		
READ Repeat	TP		X		X		X		X	X	X		
READ Repeat with Reset	TPR		X		X		X		X	X	X		
READ Buffer	TB												
READ Skip	TS	X	X	X	X	X	X	X	X	X	X	X	X
WRITE Initial	TI	X	X	X	X	X	X	X	X	X	X		
WRITE Initial with Reset	TIR	X	X	X	X	X	X	X	X	X	X		
WRITE Initial Optical	TIO									X	X		
WRITE Invitational Optical	TCO									X	X		
WRITE Continue	TT		X		X		X		X	X	X	X	X
WRITE Continue with Reset	TTR		X		X		X		X	X	X		
WRITE Continue Conversational	TTV		X		X							X	X
WRITE Conversational	TV		X	X	X	X	X			X	X	X	X
WRITE Conversational with Reset	TVR		X	X	X	X	X			X	X		
WRITE Conversational Optical	TVO									X	X		
WRITE at Line Address	TL												
WRITE at Line Address with Reset	TLR												
WRITE Erase	TS												
WRITE Erase with Reset	TSR												
WRITE Break	TB												
WRITE Positive Acknowledgment	TA		X		X		X		X	X	X		
WRITE Negative Acknowledgment	TN		X	X	X	X	X		X	X	X		
WRITE Disconnect	TN												X

*C Checking feature
D Dial-up feature
T Transmit Control feature
S Station Control feature
O Optical Image Unit feature
NS Nonswitched
SW Switched

Note: Options for which Auto Poll channel programs are generated in the IODEVICE system generation macro instruction for the line specified FEATURE = AUTOPOLL

Table 5. READ and WRITE Options for Start-Stop

Table 6. READ and WRITE Options for BSC

OPTION	TYPE CODE	Nonswitched Point-to-Point	Multipoint	Switched Point-to-Point
READ Initial	TI	X	X	X
READ Connect	TC			X
READ Connect with Tone	TCW			X
READ Continue	TT	X	X	X
READ Continue with Leading Graphics [1,4,8]	TTL	X	X	X
READ Repeat	TP	X	X	X
READ Repeat with Leading Graphics [1,4,8]	TPL	X	X	X
READ Initial Inquiry	TIQ	X		
READ Inquiry	TQ	X	X	X
READ Interrupt 7	TRV	X	X	X
WRITE Initial [2]	TI	X	X	X
WRITE Initial and Reset [2]	TIR	X	X	
WRITE Continue [2]	TT	X	X	X
WRITE Continue and Reset [2]	TTR	X	X	
WRITE Reset	TR	X	X	X
WRITE Inquiry	TQ	X	X	X
WRITE Disconnect	TD			X
WRITE Wait Before Transmit [1]	TW	X	X	X
WRITE Initial Conversational [2,5]	TIV	X	X	X
WRITE Continue Conversational [2,5]	TTV	X	X	X
WRITE Initial Transparent [3,9]	TIX	X	X	X
WRITE Initial Transparent and Reset [3,9]	TIXR	X	X	
WRITE Initial Transparent Block [9]	TIE	X	X	X
WRITE Continue Transparent [3,9]	TTX	X	X	X
WRITE Continue Transparent and Reset [3,9]	TTXR	X	X	
WRITE Continue Transparent Block [9]	TTE	X	X	X
WRITE Initial Conversational Transparent [3,6,9]	TIVX	X	X	X
WRITE Continue Converstational Transparent [3,6,9]	TTVX	X	X	X
WRITE Break	TB			X
WRITE Connect	TC			X

[1] This macro cannot be used for a 2780 with which the cental computer communicates using 6-bit Transcode.
[2] This macro cannot be used for a 2715 because text transmission to this type of station is always in transparent mode.
[3] This macro cannot be used for a 2972 because text transmission to this type of station is always in nontransparent mode.
[4] The 1800, 2715, and 2770 ignore leading graphics characters sent to them. That is, these characters are neither received into core storage (2715) or terminal buffer, nor passed to any output device attached to the station.
[5] The 1800, 2770 and 2972 } do not transmit text as a response to text received from the central computer, the return
[6] The 1800, 2715 and 2770 } the usual alternating acknowledgment (ACK-0 or ACK-1).
[7] When this macro is used for the remote 3270, the response is always EOT.
[8] This macro is not applicable for the remote 3270, which cannot receive leading graphics.
[9] This macro is not applicable for the remote 3270, because nontransparent mode is always used.

Table 6. READ and WRITE Options for BSC

OPTION	TYPE CODE
READ Initial	TI
READ Modified	TM
READ Modified from Position	TMP
READ Buffer	TB
READ Buffer from Position	TBP
WRITE Initial	TI
WRITE Erase	TS
WRITE Unprotected Erase	TUS

Table 6A. READ and WRITE Options for Local 3270 Display System

Name	Operation	Operands
[symbol]	{READ / WRITE}	decbaddr,optype,dcbaddr, {[inoutarea] / ([inarea],[outarea])} , {[inoutlength] / ([inlength],[outlength])} , [entry],[rln] [,MF=L / ,MF=E]

decbaddr
Specifies the address of the DECB associated with the line. You can use register notation only if the macro is of the execute form (MF=E).

optype
Specifies one of the operation-type codes listed in Tables 5, 6, and 6A. The channel program generated for each type of Read and Write differs depending on the particular terminal and network configuration. The available types for a given type of remote station or line configuration are given in the next two chapters. The available types for the local 3270 display system are given in the section "Local Read and Write Operations." In all cases, if the single letter T is coded, no type code is set in the DECB. The T can be used:

1. with a list form, to create a DECB with no type code. The type code would be furnished by a subsequent READ or WRITE macro of execute form.

2. with an execute form, when the type code already in the DECB is to be used.

dcbaddr
Specifies the address of the DCB for the line group.

inoutarea
Specifies the address of the first byte of the input area (Read operations) or the first byte of the output area (Write operations). In a READ macro, you may code this operand as 'S' if you are using dynamic buffering and wish BTAM to provide the needed buffers. This operand may be omitted for READ types TIQ and TQ and WRITE types TR, TW, TN, TA, TB, TD, and TW.

For WRITE type TQ, if inoutarea is omitted or inoutlength is equal to or less than 2, the response is read into the DECRESPN field of the DECB.

inarea and outarea
Are for use in READ macros of the TCW, TTL and TPL types and WRITE macros of the TIV, TIVX, TTV, and TTVX types.
For READ TTL and TPL, outarea contains the leading-graphics characters to be sent to the remote station, and inarea receives the text from the station. For Read TCW, outarea specifies the address of the tone characters to be sent to the remote station, and inarea receives the text from the station. For WRITE TIV, TIVX, TTV, and TTVX, outarea contains the text to be sent to the remote stations, and inarea receives the text transmitted from the remote station.
For either READ or WRITE macros, you may code inarea (but not outarea) as 'S' if you are using dynamic buffering and wish BTAM to provide the needed buffers.

For more information about using the inarea and outarea operands for the remote 3270 display system, see "Read Operations" and "Write Operations" under the heading "Line Control and Message Transmission" in the section "IBM 3270 Display System - Programming Considerations."

inoutlength
Specifies the number of bytes in the input or output area defined by the inoutarea operand. In a WRITE macro, you may code this operand as 'S', to cause BTAM to obtain the buffer length from the DCB.

Note: Specify 'S' only if the last buffer to be sent is completely filled, i.e., the last byte contains the ending character. The inoutlength operand need not be coded for READ types TIQ and TQ and WRITE types TR, TN, TA, TB, TD, and TW.

For WRITE type TQ, if inoutlength is omitted or is equal to or less than 2, the response is read into the DECRESPN field of the DECB.

inlength and outlength
: are for use in the same types of macros indicated under "inarea and outarea", and specify the length of these areas. In a WRITE macro, you may code outlength as 'S', to cause BTAM to obtain the buffer length from the DCB. The same caution indicated for inoutlength applies to outlength.

Programming Notes

1. The value specified for inoutlength, inlength, or outlength must include (a) all control characters that are to be sent or received if they will be sent from or received into the area (i.e., the length should not include any control characters sent automatically by BTAM or received into other than the input area, and (b) the four-byte link field, if dynamic buffering is used.

2. Any macro of the execute form that specifies inarea and outarea rather than inoutarea must refer to a DECB that has been defined with a list or standard-form macro that also specified inarea and outarea, because the DECB required for leading-graphics and conversational operations is longer than that for operations not requiring both input and output areas.

3. If on-line testing is made available for a binary synchronous line (by coding T among the EROPT options in the DCB macro for the line group), all Read Initial macros issued for the line must specify a length of no less than 300 bytes. On-line test RFT messages may be received into this area at any time. If 'T' is coded as the optype operand in a WRITE macro of the list form (MF=L), and the '(inarea, outarea)' and '(inlength,outlength)' operands are not coded, the resultant DECB does not allow space for the DECWLNG and DECWAREA fields.

entry
: specifies the address of the terminal list or an entry therein, as follows:

1. For a nonswitched line (OPENLST, AUTOLST, or SSALST), it specifies the address of an entry within the terminal list.

2. For a nonswitched line (WRAPLST, AUTOWLST, or SSAWLST), it may specify either the address of any entry within the terminal list or 'S'.

   Note: If 'S' is specified, the system will provide the address of an entry in the polling list as follows:

   a. If the previous polling operation terminated with a negative response as a result of a RESETPL macro instruction, the address of the next entry will be provided.
   b. Otherwise, the address of the entry that was last polled will be provided.

3. For a switched line (DIALST, SWLST, BSCLST, IDLST, or WTTALST), it must specify the address of the beginning of the terminal list; it cannot be coded as 'S'.

For READ types TMP and TBP for the local 3270 display system, entry specifies the address of a four-byte area that gives the position from which the read operation is to begin. This operand may be omitted for all other READ and WRITE types for the local 3270 display system, since the operand is ignored.

rln
: specifies, in decimal the relative line number within the line group. (Range 1-255 inclusive). This value is placed in the DECRLN field of the DECB, in binary form.

For READ type TI for the local 3270 display system, rln specifies the first display station that is to be checked for an attention interruption. For all other READ and WRITE types for the local 3270 display system, this operand specifies the device from which or to which a message is to be read or written.

MF=L
: specifies that this macro instruction causes only the creation of a data event control block whose name is specified by the decbaddr operand. Specify this when you wish to create a data event control block that will be referred to subsequently by one or more READ or WRITE macro instructions (each of which will specify the MF=E operand and whose decbaddr operand will specify the address of the data event control block created by this macro).

MF=E
: Specifies that this macro instruction causes execution of the Read or Write

function, using a data event control block created by a READ or WRITE macro of the list or standard form.

Return Codes: After a READ or WRITE macro instruction, BTAM sets register 15 to zero if no error has been detected. If an abnormal condition is detected, the operation is not started and control is returned to your program at the instruction following the READ or WRITE macro instruction. A return code in register 15 indicates the error. Bits 24 through 31 will contain one of the following error codes in hexadecimal notation:

04 Busy: The specified line is busy with a previously requested Read or Write operation.

08 Invalid RLN. The relative line number specified in the operand field of the READ or WRITE macro instruction is zero or is larger than the number of lines in the line group.

0C (1) Invalid "optype" code: The READ or WRITE macro instruction specified an "optype" that is invalid for the kind of remote station for which you issued the macro.

(2) An initial-type WRITE macro (e.g., WRITE TI, TIX, TIV) erroneously specified an answering list instead of a calling list (that is, no dial digits are present in the list).

10 All skip bits on (programmed polling): The skip bit is on in all of the entries in the polling or addressing list.

Usage Count too large (Auto Poll): The Usage Count is larger than its maximum value of 15.

14 Line Error during Open: SAD or Enable command (issued by Open) resulted in a permanent I/O error. The error status in the DECB may be inspected to determine the cause of the error.

18 Buffers Not Available: The buffer pool does not contain enough buffers to satisfy the Read operation (area coded 'S').

1C No Buffer Pool: No buffer pool was defined in the DCB macro instruction or there was no indication that BTAM was to provide the pool associated with the line group prior to Open.

20 No Buffer Routine: You did not indicate you wanted the buffer routine prior to OPEN, so it was not loaded with the system. The return code occurs on a Read operation.

24 Invalid Order: The second byte of the area specified by the entry operand of a READ TMP or TBP macro instruction (for a local 3270 display system) is not an SBA order.

28 Invalid Control Block: An invalid control block was encountered during a read or write operation for the local 3270 display system.

2C Device Not Available: A request for a read or write operation was rejected, because OLTEP is using the local 3270 device.

Note: All nonzero return codes indicate that no I/O operation was initiated; therefore, the program must not issue a WAIT or TWAIT macro instruction for a READ or WRITE macro instruction that resulted in a nonzero return code (the task would enter permanent wait state).

Programming Note: Execution of a READ or WRITE macro instruction causes control to be passed to a BTAM routine which constructs channel programs. If no invalid conditions are detected, a channel program will be generated for the requested I/O operation. Once the channel program has been started, control will be returned to your program with a return code of zero. The I/O operation proceeds asynchronously with respect to program execution. When you wish to determine whether the I/O operation has completed, issue a WAIT or TWAIT macro instruction, or check the DECSDECB field (the event control block).

If you intend to make use of the 'S' option for the entry operand in READ Initial (TI) macro instructions, ensure that the polling list address is placed in the data event control block before the first execution of the READ Initial (TI) macro instruction. This may be done by defining the polling list address in a READ macro instruction of the list form and then using the 'S' in a READ Initial of the execute form. However, if a WRITE Initial is issued (using the same DECB) before the first READ Initial, steps must be taken to replace the polling list address in the DECB. After the first READ Initial, BTAM maintains the polling restart address in the DECB for the line. Thus, by using the

'S' option, the polling list address is preserved across write operations (even though they utilize the same DECB field for addressing list pointers).

When a READ or WRITE macro instruction specifies "reset at completion" for a switched line, BTAM disconnects the line only if no error condition occurs during the execution of the basic channel program. Thus, the program may attempt retransmission without re-establishing the line connection. If the program elects not to attempt retransmission, the WRITE (TN) macro instruction may be executed to perform the disconnect function.

When a READ or WRITE macro instruction specifies "reset at completion" for a nonswitched line, the EOT character (or sequence of characters) will be transmitted only if no error condition occurs during execution of the basic channel program.

When a polling function is performed in a Read Initial operation, the terminal list address field in the DECB (DECENTRY) contains the address of the entry in the polling list that was last polled. Thus, the program may determine the source of the message (if one was received) by inspecting the contents of the polling list entry at that address. Note that the terminal list address field in the DECB is not modified for addressing operations.

With Auto Poll an index byte is provided in the first byte of DECPOLPT. The index byte contains the number of the polling entry for the terminal from which the message was read; that is, it contains one for the first entry in the polling list, two for the second entry, etc.). The program may obtain the index byte from DECPOLPT to identify the originating terminal following any Read operation.

When a READ macro instruction is used for the local 3270 display system, the relative line number of the device from which the message is read is placed into the DECPOLPT field of the DECB.

User program error routines that operate synchronously with respect to the completion of the I/O operation may retry a macro instruction with the knowledge that the proper parameters will be in the DECB. The error routine may use a READ or WRITE macro instruction of the execute form with only the decbaddr and the optype operands specified.

RESETPL (Reset Polling List or Reset Line) Macro Instruction

RESETPL may be issued whenever you wish to cancel a Read operation that is currently in progress but has not yet received a positive response to polling from a remote station (nonswitched multipoint line), or has not yet received an ENQ character indicating the remote station's intention to transmit (nonswitched point-to-point line), or has not yet received a call from a remote station (switched line). If at the time the RESETPL is issued a positive response or a call has been received or message transmission has taken place, the reset function has no effect; the Read operation proceeds as usual.

RESETPL functions with a nonswitched multipoint line in the following manner. If a programmed polling operation is currently in progress, and if it elicits a negative response, polling is terminated, the polling list pointer (DECPOLPT) is incremented, and the operation is posted complete. If an Auto Poll polling operation is currently in progress, and if it elicits negative responses to all entries in the list, the operation is posted complete and the index byte identifying the last active entry is stored in the first byte of DECPOLPT. In both cases (programmed and Auto Poll), the negative response bit is set in DECFLAGS. If the polling operation elicits a positive response or a time-out, the polling list pointer is not incremented, and the operation proceeds to its normal conclusion (normal conclusion for a time-out is to post it complete-with-error).

For a nonswitched point-to-point line, if a Prepare operation is currently in progress, and the Prepare has not been completed, a Halt I/O command is issued for that line. If an operation other than the Prepare is currently in progress (e.g., message reception, message transmission, addressing), it will proceed to its normal completion.

The RESETPL macro instruction functions with a switched line in the following manner. If an Enable command has been issued to a line (to allow a terminal to dial the computer), and a call has not been received (the Enable has not been completed), BTAM stops line activity by means of a Halt I/O command. If the Enable has already been completed and a polling operation (start-stop only) is currently in progress, the function described above for programmed polling or multipoint lines will be performed. If the Enable has already been completed (and for start-stop only, no polling operation is in progress), the Read operation proceeds unaffected.

Note: A special form of the RESETPL macro instruction is used for the local 3270 display system. See "Attention Interruptions and Read Initial Operations" in the section "IBM 3270 Display System—Programming Considerations." If only the decbaddr operand is specified, the instructions that are generated include support for the local 3270 display system.

Name	Operation	Operand
[symbol]	RESETPL	decbaddr [,POLLING / ,ANSRING]

decbaddr
Specifies the address of the data event control block for the line for which the reset operation is to be performed.

POLLING
Specifies that only the instructions required to terminate polling on a nonswitched line will be generated by this macro instruction.

ANSRING
Specifies that only the instructions required to terminate an answering operation on a switched line will be generated by this macro instruction.

If no second operand is specified, instructions are generated to determine at execution time shich function is to be performed. For World Trade telegraph terminal lines, omit POLLING and ANSRING.

Programming Note: No further READ or WRITE macro should be issued for a line for which a RESETPL macro has been issued until the operation in progress has been posted complete. That is, provided the RESETPL macro instruction gave a return code of X'00' or X'04', a WAIT macro instruction should be coded between the RESETPL macro and the next READ or WRITE macro.

Return Codes: After execution of a RESETPL macro instruction, bits 24 through 31 of register 15 contain a return code indicating the status of the operation. Bits 0 through 23 will contain zeros. The code will be one of the following, in hexadecimal notation:

00 Normal Return: This code will be set if an Enable or Prepare command was outstanding and a Halt I/O instruction was successfully executed.

04 Complete: this code is set if the Enable command was already completed or the Post flag in the UCB is not on.

08 Illegal Request: this code is set if the unit control block (UCB), an internal OS/360 control block, specifies a non-teleprocessing device.

0C Unsuccessful: this code will be set if an invalid unit control block address has been passed to the IOHALT system macro instruction, or if the Halt I/O instruction has terminated in error (e.g., Channel Data Check, or a Not Operational condition code). This code will also be set if the DCB has not been opened.

10 Not Issued: This code is set if no Enable command had been issued.

Note: These return codes have different meanings for the form of the RESETPL macro instruction used for the local 3270 display system. See "Attention Handling and Read Initial Operations" in the section "IBM 3270 Display System - Programming Considerations."

WAIT Macro Instruction

The WAIT macro instruction relinquishes control of the CPU when the user program has no further processing to do and must wait for the completion of one or more Read/Write operations. See the OS Supervisor and Data Management Macro Instructions publication for complete information on this macro.

Name	Operation	Operand
[symbol]	WAIT	[count], {ECB=ecb address / ECBLIST=ecb list addr}

count
Specifies the number of events among the events referred to by the ECB or ECBLIST operand that must be posted complete before the WAIT macro is satisfied. If the count operand is omitted, 1 is assumed.

ECB
Specifies the address of an event control block (ECB) representing a single event to be posted complete before processing by the user program can continue.

ECBLIST
Specifies the address of a variable-length list containing fullword entries with each fullword entry containing the address of an event control block (ECB) in the low-order three bytes. Each event control block pointed to represents an event awaiting completion. In this list of ECB addresses, the high-order bit (0-bit) of each fullword entry except the last in the list must be zero. In the last entry in the list, you must set the 0-bit in the high-order byte of the entry to 1.

TWAIT Macro Instruction

The TWAIT macro instruction relinquishes control of the CPU when the user program must wait for the completion of one of a number of events before further processing can be done.

Name	Operation	Operands
[symbol]	TWAIT	(returnreg), ECBLIST=ecb list addr

returnreg
Contains the address of the ECB representing the event posted complete.

ECBLIST
Specifies the address of the user-created list of ECB addresses representing events awaiting completion. Each entry in the list is a fullword containing an address in the low-order three bytes. Because the list is of variable length, the high-order bit (0-bit) of each fullword entry (except the last) must be set to zero except that of the last entry. The high-order bit of the last fullword entry must be set to one to identify the entry as the last in the list.

If TWAIT is issued for any event other than a Read or Write operation, your program should clear the ECB. The TWAIT macro instruction is similar to the WAIT macro instruction except that:

- TWAIT requires the completion of one event, only, before returning control to the problem.
- The ECB keyword is not used in TWAIT.
- The address of the ECB which was posted complete is returned to you in the register specified (as the first operand of TWAIT).
- The displacement of the ECB address from the beginning of the ECB list (as specified in the ECBLIST operand) is returned in register 15.

The last point above simplifies branching to a routine associated with a particular ECB. Set up a where-to-go list of four-byte entries, each entry containing the address of a routine to be associated with the ECB whose address is in the corresponding entry of the ECB list. Then you may code, following the TWAIT macro,

```
L        15,WTGLIST(15)
BALR     14,15
```

or the equivalent, to branch and link to the appropriate routine.

Read Skip Operations

One of the types of Read operations that may be performed for terminals on a start-stop communications line is Read Skip. In this operation, effected by the READ Skip (TS) macro instruction, any data being received from a terminal is discarded, instead of being placed in main storage. This action "clears the line" of any unwanted data, so that normal Read and Write operations can be resumed. A Read Skip operation is intended for use in user-written recovery routines when a lost-data error has occurred. (It may be used for any start-stop terminal.)

Because it is not used in normal message transmission operations, Read Skip is not discussed in the Start-Stop Read and Write Operations chapter.

In a READ Skip macro you need not specify the address of an input area, since the data it receives does not enter main storage; however, you must specify for the length a value exceeding the maximum amount of text data to be received by the Read Skip operation.

IBM 1030 DATA COLLECTION SYSTEM

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an open or wraparound polling list. The list may have one or more terminal entries, each containing a single polling character that identifies the terminal. To define a polling list, code the operand field of a DFTRMLST macro like this:

```
{OPENLST}
{WRAPLST},(xx,...)
```

Write Operations

A Write Initial operation requires an addressing list having a single entry, containing a single addressing character that identifies the terminal that is to receive the output message. To define an addressing list, code the operand field of a DFTRMLST macro like this:

```
OPENLST,xx
```

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)

READ Initial successively polls the terminals in the polling list, and upon receiving a positive response to polling, reads a message block.

1. Write EOT EOT EOT
2. Write Polling Character
3. Read Response
4. Read Text
5. Write Y EOT EOT EOT (TIR only)

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response, successively polls terminals in the polling list, beginning with the terminal to which it sent the response, and upon receiving a positive response to polling reads a message block. This macro is for use following a successful READ Initial (TI) or another READ Continue to receive another message block.

1. Write Y EOT EOT EOT
2. Write Polling Character
3. Read Response
4. Read Text
5. Write Y EOT EOT EOT (TTR only)

READ Repeat (TP)
READ Repeat and Reset (TPR)

READ Repeat writes a negative response, successively polls the terminals in the polling list, beginning with the terminal to which it sent the negative response, and upon receiving a positive response to polling, reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT) or READ Repeat (TP), to read the same message block received by the previous operation.

1. Write N EOT EOT EOT
2. Write Polling Character
3. Read Response
4. Read Text
5. Write Y EOT EOT EOT (TPR Only)

WRITE MACRO INSTRUCTIONS

Programming Notes:

1. If an EOA is the first character of a message block (as it will be if the block has not been modified since it was received from a 1030 terminal), it will print at the terminal as #. You should therefore overlay with an idle character the first character of each block received from a 1030.

2. Each outgoing message block must end with ETB.

3. Insert three idle (EBCDIC) or Write Marks (transmission code) characters between adjacent message text characters. (This is required of all messages sent to a 1033 printer.)

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial addresses a terminal, writes an EOA to place the terminal in receive state, writes message text, and reads a response from the terminal.

1.	Write	EOT EOT EOT S
2.	Write	Addressing Character
3.	Write	"1"
4.	Read	Response
5.	Write	EOA
6.	Write	Text
7.	Read	Response
8.	Write	EOT EOT EOT (TIR only)

WRITE Continue (TT)
WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1.	Write	Text
2.	Read	Response
3.	Write	EOT EOT EOT (TTR only)

WRITE Positive Acknowledgment (TA)

WRITE Positive Acknowledgment writes a positive acknowledgment and an EOT sequence to indicate to the terminal that the computer received message text without error and to stop line activity. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write Y EOT EOT EOT

WRITE Negative Acknowledgment (TN)

WRITE Negative Acknowledgment writes an EOT sequence to indicate to the terminal that the computer received text with an error and to stop line activity. The terminal interprets the EOT sequence as a negative response. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write EOT EOT EOT

IBM 1050 DATA COMMUNICATIONS SYSTEM -- NONSWITCHED LINES

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an open or wraparound polling list. The list may have one or more terminal entries, each containing a two-character polling sequence. The first character identifies the terminal; the second identifies the specific component from which an input message is solicited. (If the second character is the common polling character, 0, input messages are read from any ready component. To define a polling list, code the operand field of a DFTRMLST macro like this:

```
{OPENLST}
{WRAPLST},(xxyy,...)
```

Write Operations

A Write Initial operation requires an addressing list having one or more terminal entries, each containing a two-character addressing sequence. The first character identifies the terminal; the second identifies the specific component that is to receive the output message. (If the second character is the common addressing character, 9, the output message is sent to all ready components.) To define an addressing list, code the operand field of a DFTRMLST macro like this:

```
OPENLST,(xxyy,...)
```

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)

READ Initial successively polls the terminals in the polling list, and upon receiving a positive response to polling, reads a message block.

1. Write EOT EOT EOT
2. Write Polling Sequence
3. Read Response
4. Read Text
5. Write EOA EOT EOT EOT (TIR only)

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same terminal and component that sent the previous block.

1. Write Y
2. Read Text
3. Write EOA EOT EOT EOT (TIR only)

READ Repeat (TP)
READ Repeat and Reset (TPR)

READ Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1. Write N
2. Read Text
3. Write EOA EOT EOT EOT (TPR only)

WRITE MACRO INSTRUCTIONS

Programming Notes:

1. The first block of a message received from a 1050 on a Read Initial operation will begin with an EOA character. If the same message block is then sent to a 1050, it will be printed as #. This may be avoided by overlaying the EOA with an Idle character before sending the message block.
2. Each outgoing message block must end with EOB.

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial addresses a terminal, and if the response to addressing is positive, writes an EOA followed by message text and reads the response to text. If the terminal sends a negative response to addressing, the operation is posted complete.

1. Write EOT EOT EOT
2. Write Addressing Sequence
3. Read Response
4. Write EOA

5. Write Text
6. Read Response
7. Write EOT EOT EOT (TIR only)

WRITE Continue (TT)
WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1. Write Text
2. Read Response
3. Write EOT EOT EOT (TTR only)

WRITE Continue Conversational (TTV)

WRITE Continue Conversational writes a message block and reads a response from the terminal, then resets the terminals on the line to control mode, successively polls the terminals in the polling list, and upon receiving a positive response to polling, reads a message block.

1. Write Text
2. Read Response
3. Write EOT EOT EOT
4. Write Polling Sequence
5. Read Response
6. Read Text

Programming Notes:

1. WRITE TTV performs exactly the same functions as would be performed by a WRITE Continue (TT) followed by a READ Initial (TI), but saves coding effort by allowing you to verify successful initiation and conclusion of the operation (i.e., by checking return and completion codes) just once, instead of after each of the two separate macro instructions.

2. You must specify a polling list entry in the WRITE TTV macro instruction.

WRITE Positive Acknowledgment (TA)

WRITE Positive Acknowledgment writes a positive acknowledgment and an EOT sequence to indicate to the terminal that the computer received message text without error and to stop line activity. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write EOA EOT EOT EOT

WRITE Negative Acknowledgment (TN)

WRITE Negative Acknowledgment writes an EOT sequence to indicate to the terminal that the computer received text with an error and to stop line activity. The terminal interprets the EOT sequence as a negative response. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write EOT EOT EOT

IBM 1050 DATA COMMUNICATIONS SYSTEM -- SWITCHED LINES

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation that answers a call from a terminal requires an answering-polling list; a Read Initial operation that calls a terminal requires a calling-polling list. Either type of list may have one or more terminal entries (all representing the same terminal), each containing a two-character polling sequence. The first character identifies the terminal, and must be the same character for all entries in the list; the second character identifies the specific component from which an input message is solicited. (If the second character is the common polling character, 0, input messages are read from any ready component.)

To define an answering-polling list, code the operand field of a DFTRMLST macro like this:

```
DIALST,0,(xxyy,...)
```

To define a calling-polling list, code the DFTRMLST operand field like this:

```
DIALST,dialcount,dialchars,(xxyy,...)
```

Write Operations

A Write Initial operation that calls a terminal requires a calling-addressing list; a Write Initial operation that answers a call from a terminal requires an answering-addressing list. Either type of list may have one or more terminal entries (all representing the same terminal), each containing a two-character addressing sequence. The first character identifies the terminal, and must be the same for all entries in the list; the second character identifies the specific component that is to receive the output message. (If the second character is the common addressing character, 9, the output message is sent to all ready components.)

To define a calling-addressing list, code the operand field of a DFTRMLST macro like this:

```
DIALST,dialcount,dialchars,(xxyy,...)
```

To define an answering-addressing list, code the operand field of a DFTRMLST macro like this:

```
DIALST,0,(xxyy,...)
```

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)
(Using Answering-Polling List)

READ Initial answers a call from a terminal, polls it, reads the response, and if the response is positive, reads a message block. If the response is negative, the operation is posted complete.

1. Disable
2. Enable
3. Write Pad characters
4. Write EOT EOT EOT
5. Write Polling sequence
6. Read Response
7. Read Text
8. Write EOA EOT (TIR only)
9. Disable (TIR only)

READ Initial (TI)
READ Initial and Reset(TIR)
(Using Calling-Polling List)

READ Initial dials the terminal, polls it, and if the response is positive, reads a message block. If the response is negative, the operation is posted complete.

1. Disable
2. Dial Dial Digits
3. Write Pad characters
4. Write EOT EOT EOT
5. Write Polling sequence
6. Read Response
7. Read Text
8. Write EOA EOT (TIR only)
9. Disable (TIR only)

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same terminal and component that sent the previous block.

1. Write Y
2. Read Text
3. Write EOA EOT (TTR only)
4. Disable (TTR only)

READ Repeat (TP)
READ Repeat and Reset (TPR)

READ Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1. Write N
2. Read Text
3. Write EOA EOT EOT EOT (TPR only)
4. Disable (TPR only)

Programming Note: In order to be able to issue READ Repeat for the paper tape reader or card reader, the reader must be equipped with the Line Correction feature. For either of these components you may issue it only twice in succession. Furthermore, you should use this macro for the paper tape reader only if the message block being read is less than 312 characters. (Otherwise, the time required to back up the tape for retransmission exceeds the time-out interval of the terminal.)

READ Conversational (TV)
READ Conversational and Reset (TVR)

READ Conversational polls the terminal, and if the response to polling is positive, reads a message block. If the response is negative, the operation is posted complete. This macro is for polling and reading a message block from a terminal with which a previous READ or WRITE macro has already established the line connection. Its main purpose is to allow you to change from sending message blocks to receiving them, without having to reestablish the line connection. READ Conversational can follow a READ Initial (TI), READ Continue(TT), or READ Repeat (TP), or a WRITE Initial (TI), WRITE Continue (TT), or WRITE Conversational (TV).

The terminal list used by the READ Conversation must be an open polling list (calling or answering), of the DIALST format.

1. Write EOT EOT EOT
2. Write Polling Sequence
3. Read Response
4. Read Text
5. Write EOA EOT (TVR only)
6. Disable (TVR only)

WRITE MACRO INSTRUCTIONS

Programming Notes:

1. The first block of a message received from a 1050 on a Read Initial or Read Conversational operation will begin with an EOA character. If the same message block is then sent to a 1050, it will print as #. This may be avoided by overlaying the EOA with an Idle character before sending the block.

2. Each outgoing message block must end with EOB.

WRITE Initial (TI)
WRITE Initial and Reset (TIR)
(Using Calling-Addressing List)

WRITE Initial dials a terminal, addresses it, and if the response to addressing is positive, writes an EOA followed by message text and reads the response to text. If the response to addressing is negative, the operation is posted complete.

1. Disable
2. Dial Dial digits
3. Write Pad characters
4. Write EOT EOT EOT
5. Write Addressing sequence
6. Read Response
7. Write EOA
8. Write Text
9. Read Response
10. Write EOT (TIR only)
11. Disable (TIR only)

WRITE Initial (TI)
WRITE Initial and Reset (TIR)
(Using Answering-Addressing List)

WRITE Initial answers a call from a terminal, addresses it, and if the response is positive, writes an EOA followed by message text and reads the response to text. If the response to addressing is negative, the operation is posted complete.

1. Disable
2. Enable

3. Write Pad characters
4. Write EOT EOT EOT
5. Write Addressing sequence
6. Read Response
7. Write EOA
8. Write Text
9. Read Response
10. Write EOT (TIR only)
11. Disable (TIR only)

WRITE Continue (TT)
WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1. Write Text
2. Read Response
3. Write EOT (TTR only)
4. Disable (TTR only)

WRITE Conversational (TV)
WRITE Conversational and Reset (TVR)

WRITE Conversational writes a positive response to text (the EOA character is the positive response), addresses the terminal, and if the response to addressing is positive, writes an EOA followed by message text and reads the response to text. If the response to addressing is negative, the operation is posted complete.

The terminal list used by the WRITE Conversational must be an addressing list (calling or answering) of the DIALST format.

1. Write EOA EOT EOT EOT
2. Write Addressing sequence
3. Read Response
4. Write EOA
5. Write Text
6. Read Response
7. Write EOT (TVR only)
8. Disable (TVR only)

WRITE Continue Conversational (TTV)

WRITE Continue Conversational writes a message block and reads a response from the terminal, resets it to control mode, polls it, and upon receiving a positive response to polling, reads a message block.

1. Write Text
2. Read Response
3. Write EOT EOT EOT
4. Write Polling Sequence
5. Read Response
6. Read Text

Programming Notes:

1. WRITE TTV performs exactly the same functions as would be performed by a WRITE Continue (TT) followed by a READ Conversational (TV), but saves coding effort by allowing you to verify successful initiation and conclusion of the operation (i.e., by checking return and completion codes) just once, instead of after each of the two separate macro instructions.

2. You must specify a polling list entry in the WRITE TTV macro instruction.

WRITE Positive Acknowledgment and Disconnect (TA)

WRITE Positive Acknowledgment and Disconnect writes a positive response to text (an EOA) and breaks the line connection. This macro is for use following a successful READ operation when you wish to break the line connection instead of receiving the remaining blocks of a message.

1. Write EOA EOT
2. Disable

WRITE Negative Acknowledgment and Disconnect (TN)

WRITE Negative Acknowledgment and Disconnect writes a negative acknowledgment (the EOT character serves this purpose) and breaks the line connection. This macro is for use following an unsuccessful Read operation when you wish to break the line connection instead of receiving the remaining blocks of a message. The macro may also be used after a write operation when you wish to break the line connection.

1. Write EOT
2. Disable

IBM 1060 DATA COMMUNICATION SYSTEM

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an open or wraparound polling list. The list may have one or more terminal entries, each containing a two- character polling sequence. The first character identifies the control unit, the second identifies the teller terminal that is to be polled.

To define a polling list, code the operand field of a DFTRMLST macro like this:

```
{OPENLST}
{WRAPLST},(xxyy,...)
```

Write Operations

A Write Initial operation requires an addressing list having one terminal entry that contains a two-character addressing sequence. The first character identifies the control unit, the second identifies the teller terminal to which the message is to be sent.

To define an addressing list, code the DFTRMLST operand field like this:

```
OPENLST,xxyy
```

The list must be defined as an open list.

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)

READ Initial successively polls the terminals in the polling list, and upon receiving a positive response to polling, reads a message block.

1.	Write	EOT EOT EOT
2.	Write	Polling Sequence
3.	Read	Response
4.	Read	Text
5.	Write	Y EOT EOT EOT (TIR only)

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response, successively polls terminals in the polling list, beginning with the terminal to which it sent the response, and upon receiving a positive response to polling, reads a message block. This macro is for use following a successful READ Initial (TI) or another READ Continue to receive another message block.

1.	Write	Y EOT EOT EOT
2.	Write	Polling Sequence
3.	Read	Response
4.	Read	Text
5.	Write	Y EOT EOT EOT (TTR only)

READ Repeat (TP)
READ Repeat and Reset (TPR)

READ Repeat writes a negative response, successively polls the terminals in the polling list, beginning with the terminal to which it sent the negative response, and upon receiving a positive response to polling reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT) or READ Repeat (TP), to read the same message block received by the previous operation.

1.	Write	N EOT EOT EOT
2.	Write	Polling Sequence
3.	Read	Response
4.	Read	Text
5.	Write	Y EOT EOT EOT (TPR only)

WRITE MACRO INSTRUCTIONS

Programming Note: Each outgoing message block must end with ETB.

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial addresses a terminal, and if the response to address is positive, writes an EOA followed by message text and reads the response to text. If the terminal sends a negative response to addressing, the operation is posted complete.

1.	Write	EOT EOT EOT
2.	Write	Addressing sequence
3.	Read	Response
4.	Write	EOA
5.	Write	Text

6. Read Response
7. Write EOT EOT EOT (TIR only)

WRITE Positive Acknowledgment (TA)

WRITE Positive Acknowledgment writes a positive acknowledgment and an EOT sequence to indicate to the terminal that the computer received message text without error and to stop line activity. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write Y EOT EOT EOT

WRITE Negative Acknowledgment (TN)

WRITE Negative Acknowledgment writes an EOT sequence to indicate to the terminal that the computer received text with an error and to stop line activity. The terminal interprets the EOT sequence as a negative response. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write EOT EOT EOT

IBM 2260-2848 DISPLAY COMPLEX (REMOTE)
IBM 2265-2845 DISPLAY COMPLEX (REMOTE)

The information in this section applies equally to the IBM 2260-2848 display complex and the IBM 2265-2845 display complex, except that references to multiple display stations and the general polling function do not apply to the 2265-2845 display complex (only one 2265 display station can be attached to a 2845 display control).

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an open or wraparound polling list. The list may have one or more terminal entries, each containing a two-character polling sequence. The first character identifies the 2848 or 2845 Display Control, the second identifies the 2260 or 2265 Display Station from which an input message is solicited. (If, for a 2260/2848, the second character is coded as X'FF', a general poll is performed.)

To define a polling list, code the operand field of a DFTRMLST macro like this:

```
{OPENLST},(xxyy,...)
{WRAPLST}
```

Write Operations

A Write Initial operation requires an addressing list. The list may have one or more terminal entries, each containing a two-character addressing sequence. The first character identifies the 2848 or 2845 Display Control, the second identifies the 2260 or 2265 Display Station or 1053 printer that is to receive the output message.

To define an addressing list, code the DFTRMLST operand field like this:

```
OPENLST,(xxyy,...)
```

READ MACRO INSTRUCTIONS

The format of a message received through a READ macro instruction is:

STX	device address	[text]	[CAN]	ETX

STX
: specifies the start of text characters.

device address
: identifies the sending unit (display station or printer).

text
: is the message text.

CAN
: is the cancel character, sent only if the display control detects an internal operation error when transmitting the message.

ETX
: is the end-of-text character.

READ Initial (TI)
READ Initial and Reset (TIR)

READ Initial successively polls the display stations and printers in the polling list, and upon receiving a positive response to polling, reads a message block or a printer status message.

1. Write	STX and 15 EOT's	
2. Write	Polling Sequence	
3. Write	READ MI code	
4. Read	Response	
5. Read	Text	
6. Write	STX EOT EOT EOT	(TIR only)

Functions of the Read Initial Operation

The Read Initial operation can have one of three functions:

Specific Polling of One or More Display Stations: A Read Initial operation executed for this purpose is similar to the Read Initial for other types of terminals that use the polling scheme. That is, the stations to be polled are individually represented in a terminal list and polling proceeds until the end of the list is reached (open list) or until a station returns a positive response and a message block (open or wraparound list).

Requesting Printer Status: The function of polling ordinarily refers to contacting a terminal or terminal component to determine whether it has any message to send to the computer. With reference to the 1053 printer attached to an IBM 2848 Display Control, the term polling means contacting the printer to see if it is ready to receive a message from the computer. (The printer may not be ready because the terminal operator is using it locally or because its power is off or it is out of paper.) In order to be ready, the printer mechanism must be ready and the printer buffer must not be in use. (The printer buffer is in use when it is being filled from the keyboard buffer or the line, or if its contents are currently being printed.) If the printer, when polled, is ready, it returns to the computer a positive response, the format of which is STX (device address) ETX.

If the printer is not ready, it returns a NAK; if the printer mechanism is ready but the printer buffer is in use, the printer returns an EOT. Either NAK or EOT is considered a negative response. Once the printer status is requested, whether the status is positive or negative the printer is in the "printer-request" condition, which means that it is available only for a message sent by a Write operation.

If a general poll operation (see below) is being executed after the status is requested, the printer returns the positive response when its status changes from not-ready to ready. The response is given the first time the printer is polled following the change to the ready condition.

General Polling of a 2848 Display Control: In this form of operation, which is achieved by coding X'FF' as the second byte of a single polling list entry, all the display stations and printers connected to the display control identified by the first character of the entry are polled in wrap-around fashion, i.e., continuously, until one of the stations returns a positive response or until the printer responds with a status indication. The printer is always polled first, then the display stations, in each polling "pass". (The printer status is returned when it is polled, either if a previous Read Initial operation requested the status indication or if a previous Write Initial operation for the printer was not possible because the printer was not free.)

When a display station sends a response, the Read operation receives the message block into the input area specified in the READ macro. When a printer returns a response, the Read operation receives the status indication in the input area.

Programming Note: It is advisable to turn on the end-of-list bit of the polling list entry for the printer when executing a Read Initial operation to request printer status; then if the printer returns a negative response the operation is posted complete and the negative response condition can be tested for in the DECB. If the end-of-list bit is not on, the negative response is treated just like a negative response from a display station; polling continues, with no response indication given in the DECB.

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same display station that sent the previous block.

1. Write ACK
2. Read Text
3. Write STX EOT (TTR only)

READ Repeat (TPR)
READ Repeat and Reset (TPR)
(Display Only)

Read Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1. Write NAK
2. Read Text
3. Write STX EOT (TPR only)

READ Buffer (TB)
READ Buffer and Reset (TBR)
(Display only)

READ Buffer is intended for special applications and for use in diagnosing equipment troubles. It receives the entire contents of the buffer of the specified display station.

1. Write STX and 15 EOT's
2. Write Polling sequence
3. Write Read Buffer Code
4. Read Response
5. Read Text
6. Write STX EOT EOT EOT (TBR only)

Programming Note: At the completion of the Read Buffer operation you must issue a WRITE Erase to erase the screen or else write a message that will overlay the previous buffer.

WRITE MACRO INSTRUCTIONS

The format of a message sent through a WRITE macro instruction is:

STX	text	ETX

The ETX character must be the last character of a message.

WRITE Initial (TI)
WRITE Initial and Reset (TIR)
(Display or Printer)

WRITE Initial addresses a display station or printer, and if the response to addressing is positive, writes an STX and the message text, then reads the response from the addressed unit. If the response to addressing is negative, the operation is posted complete.

1. Write STX and 15 EOT's
2. Write Addressing sequence
3. Write Write Code
4. Read Response
5. Write STX
6. Write Text
7. Read Response
8. Write STX EOT EOT EOT (TIR only)

Programming Notes:

1. If you issue a WRITE Initial to send a message block to the printer and the printer returns a negative response (NAK or EOT) to addressing, the operation is posted complete; the message text is not sent. If the response is positive, the message is sent. If an error occurs during transmission of text, the printer buffer is cleared. You may retry the operation with a WRITE Continue macro.

2. If you issue a WRITE Initial to send a message block to a display station, a positive response is normally received and the text is sent. If during transmission of text, an error occurs, you may retry the operation by issuing a WRITE Continue, but the message containing the error is not cleared. You may also resend the message with a WRITE Erase macro, or with a READ Buffer followed by a WRITE Erase, if several messages were displayed.

WRITE at Line Address (TL)
WRITE at Line Address and Reset (TLR)
(Display Only)

WRITE at Line Address has the same function as a WRITE Initial, but permits specifying the display line on which the message is to be displayed.

1. Write STX and 15 EOT's
2. Write Addressing sequence
3. Write Line Address Code
4. Read Response
5. Write STX
6. Write Text
7. Read Response
8. Write STX EOT EOT EOT (TLR only)

Programming Note: The first byte of the message text must contain a line address character. The message will be displayed starting at the beginning of that line. See Table 7 for valid display line addresses.

Table 7. IBM 2260 and 2265 Display Line Addresses

	Display Line Addresses		
		EBCDIC	
Display Line	2260 or 2265 Code (hex)	Code(hex)	Character
1	50	F0	0
2	51	F1	1
3	52	F2	2
4	53	F3	3
5	54	F4	4
6	55	F5	5
7	56	F6	6
8	57	F7	7
9	58	F8	8
10	59	F9	9
11	5A	7A	:
12	5B	5E	;
13	5C	4C	<
14	5D	7E	=
15	5E	6E	>

WRITE Erase (TS)
WRITE Erase and Reset (TSR)
(Display Only)

WRITE Erase has the same function as a WRITE Initial, but also causes the display station screen to be erased before the message is displayed.

1. Write STX and 15 EOT's
2. Write Addressing sequence
3. Write Erase code
4. Read Response
5. Write STX
6. Write Text
7. Read Response
8. Write STX EOT EOT EOT (TSL only)

WRITE Continue (TT)
WRITE Continue and Reset (TTR)
(Display or Printer)

WRITE Continue writes a message block and receives a response. This macro is for use following any Read or Write operation that did not include the Reset function.

1. Write STX
2. Write Text
3. Read Response
4. Write STX EOT EOT EOT (TTR only)

WRITE Positive Acknowledgment (TA)
(Display Only)

WRITE Positive Acknowledgment writes a positive acknowledgment and an EOT to indicate that the computer received the message text without error and to stop line activity. This macro is for use following a Read operation when you wish to stop receiving from the display station.

1. Write STX EOT

WRITE Negative Acknowledgment (TN)
(Display Only)

WRITE Negative Acknowledgment writes an EOT, which constitutes a negative response indicating that the computer received the message text with an error and to stop line activity. This macro is for use following a Read operation when you wish to stop receiving from the display station.

1. Write EOT

IBM 2740 COMMUNICATIONS TERMINAL--GENERAL INFORMATION

The IBM 2740 Communications Terminal (Model 1 or 2) is available with several features or combinations of features. Of these, BTAM supports the following:

2740 on Nonswitched Lines:

- Basic 2740
- 2740 with Checking
- 2740 with Station Control
- 2740 with Checking and Station Control
- 2740 with Checking and 2760 Optical Image Unit

2740 on Switched Lines:

- 2740 Dial
- 2740 Dial, with Checking
- 2740 Dial, with Transmit Control
- 2740 Dial, with Checking and Transmit Control
- 2740 Dial, with Checking and 2760 Optical Image Unit

The channel programs differ for the various feature combinations and are therefore explained separately on the following pages. (See a subsequent heading, IBM 2760 Optical Image Unit, for information about the 2760.)

Programming Notes:

1. Operator awareness: The 2740 without station control and transmit control does not react to a transmission control unit (TCU) timeout, nor does it time out along with the TCU. Following a TCU timeout, the 2740 is left in transmit text mode and is unresponsive to channel commands. The terminal operator must then depress the EOT key or power down and back up to place the 2740 in control receive mode.

2. Only 2740 terminals equipped with the Checking feature provide an automatic response to messages received from the computer. For 2740s not having this feature, responses, if desired, must be entered manually from the 2740 keyboard, and the program must be able to receive such responses. That is, each Write operation for which a response is required must be followed by a Read operation. It is up to the user to establish a convention for responses; for example, to consider the letter Y received from the terminal as a positive response (the terminal received the message correctly) and the letter N received from the terminal as a negative response (the terminal received the message incorrectly and the program should resend the same message).

3. Each message sent to a 2740 Model 2 that is equipped with the Buffered Receive feature must end with an EOT character; the EOT must be supplied by the user program.

4. Multiple-block messages must not be sent to a 2740 Model 2 that is equipped with the Buffered Receive and Checking features, because (a) the contents of the buffer are printed only when an EOT is received from the computer, and (b) all blocks are read into the same buffer. This means that if a multiple-block message is received, only the block received just prior to the EOT will be printed; all previous blocks will have been successively overlaid in the buffer.

5. In sending message text to a 2740 Model 2 with the Buffered Receive feature, be careful to avoid a buffer overflow condition that will occur if the central computer sends a message block exceeding the capacity of the terminal buffer. This can happen even when the length of the message block in main storage is less than the buffer size. For each change in case (upper to lower, or vice versa), the TCU inserts a shift character in the data stream going to the terminal. You should ensure not only that the length of the message block in main storage is shorter than the 2740 buffer, but that it is shorter by an amount sufficient to allow for the inserted shift characters.

6. The Model 2 responds to addressing with a two-character reply. If the response is positive, the first character indicates whether an error occurred during the previous Write operation while transferring data from the buffer to the printer; if an error occurred, its nature is indicated. The second character is the positive response, Y.

   If the response is negative, the first character indicates the reason for that response; the second character is the negative response, N.

   The two-character response is received in the DECRESPN field of the DECB for the line.

   The operation is posted complete, with or without error, in the event control block for the line, and the appropriate bits are set in the DECFLAGS field of the DECB.

Following each Write operation you should examine the first byte of the DECRESPN field to determine whether an error occurred, and what kind it is.

The characters (in hexadecimal notation) and their meanings are as follows.

Positive response (second character is Y)

First Character	Meaning
X'01'	(No error; buffer successfully printed)
X'23'	Failure in electronic circuit
X'25'	I/O device failure
X'29'	VRC error in text received on line
X'31'	Parity error in text received on line.

When the first character is other than X'01', BTAM prints message IEA000I (I/O ERR) at the central computer console (and/or teleprocessing or other console, if the operating system includes multiple console support). See Appendix C for the format of this message.

Negative response (second character is N)

First Character	Meaning
X'04'	Terminal is in Bid mode
X'02'	Terminal is in Communicate mode
X'20'	Terminal is in Communicate mode with document device down
X'10'	Terminal is in Local mode
X'13'	Terminal is in Communicate mode but is out of paper
X'08'	Contents of buffer are being printed.

When the first character is X'10, X'13', or X'20', BTAM posts the operation complete-with-error (completion code X'41' in DECSDECB) and prints message IEA000I (I/O ERR) at the central computer console (and/or teleprocessing or other console, if the operating system includes multiple console support). See Appendix C for the format of this message. When the first character is X'02', X'04', or X'08', BTAM posts the operation with normal completion.

IBM 2740 (BASIC)

Read and Write operations for the basic 2740 require no terminal lists.

READ MACRO INSTRUCTIONS

READ Initial (TI)

READ Initial monitors the line for an EOA sent by the terminal and reads the message block that follows. This is the only macro used to receive text.

1. Write EOT
2. Prepare
3. Read Text

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial writes an EOA to place the terminal in receive state, then writes message text. This is the only macro used to send text.

1. Write EOA and 15 Idle Characters
2. Write Text
3. Write EOT (TIR only)

IBM 2740 WITH CHECKING FEATURE

The macro instructions in this section apply to a 2740 used as an operator's console (under the Multiple Console Support option of the operating system) as well as to a 2740 used as a regular terminal.

Read and Write operations for the 2740 with the checking feature require no terminal lists.

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)

READ Initial monitors the line for an EOA sent by the terminal and reads the message block that follows.

1.	Write	EOT EOT EOT
2.	Prepare	
3.	Sense	
4.	Read	Text
5.	Write	EOA EOT EOT EOT (TIR only)

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same terminal and component that sent the previous block.

1.	Write	Y
2.	Read	Text
3.	Write	EOA EOT EOT EOT (TTR only)

READ Repeat (TP)
READ Repeat and Reset (TPR)

READ Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1.	Write	N
2.	Read	Text
3.	Write	EOA EOT EOT EOT (TPR only)

WRITE MACRO INSTRUCTIONS

Programming Notes

1. Each outgoing message block must end with EOB.

2. Once it is in receive mode, the terminal cannot begin sending message text until it receives EOT. Therefore, following one or more Write operations, you must arrange to send EOT to put the terminal in stand-by mode. This may be done by specifying the reset option in the last Write operation (i.e., TIR, TTR, or TVR), or by following the last Write operation by a Write TN macro.

Restriction:
If a Read Initial operation immediately follows a Write with Reset operation, the first byte of data may be lost.

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial writes an EOA to place the terminal in receive state and turn on the terminal motors, writes message text, and reads the response.

1.	Write	EOA and 15 idle Characters
2.	Write	Text
3.	Read	Response
4.	Write	EOT (TIR only)

WRITE Continue (TT)
WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1.	Write	Text
2.	Read	Response
3.	Write	EOT (TTR only)

WRITE Conversational (TV)
WRITE Conversational and Reset (TVR)

WRITE Conversational writes an EOA to place the terminal in receive state, writes message text, and reads the response. This macro is for use following a Read operation, to change from receiving text to sending text.

1. Write EOA
2. Write Text
3. Read Response
4. Write EOT (TVR only)

WRITE Continue Conversational (TTV)

WRITE Continue Conversational writes a message block and reads a response from the terminal, then resets it to control mode, monitors the line for an EOA from the terminal and reads the message block that follows.

1. Write Text
2. Read Response
3. Write EOT EOT EOT
4. Prepare
5. Sense
6. Read Text

Programming Note: WRITE TTV performs exactly the same functions as would be performed by a WRITE Continue (TT) followed by a READ Initial (TI), but saves coding effort by allowing you to verify successful initiation and conclusion of the operation (i.e., by checking return and completion codes) just once, instead of after each of the two separate macro instructions.

WRITE Positive Acknowledgment (TA)

WRITE Positive Acknowledgment writes a positive acknowledgment and an EOT sequence to indicate to the terminal that the computer received message text without error and to stop line activity. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write EOA EOT EOT EOT

WRITE Negative Acknowledgment (TN)

WRITE Negative Acknowledgment writes an EOT sequence to indicate to the terminal that the computer received text with an error and to stop line activity. The terminal interprets the EOT sequence as a negative response. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT, or after one or more Write operations, when you wish to begin receiving from the terminal via Read operations (the terminal cannot begin sending text until it receives EOT).

1. Write EOT EOT EOT

IBM 2740 WITH DIAL-UP FEATURE

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an answering list, which you define by coding the operand field of a DFTRMLST macro like this:

```
DIALST,0
```

Write Operations

A Write Initial operation requires a calling list, which you define by coding the DFTRMLST operand field like this:

```
DIALST,dialcount,dialchars
```

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)

READ Initial answers a call from a terminal and reads a message block.

1. Disable
2. Enable
3. Prepare
4. Read Text
5. Write EOT (TIR only)
6. Disable (TIR only)

READ Conversational (TV)
READ Conversational and Reset (TVR)

READ Conversational monitors the line for an EOA sent by the terminal and reads the message text that follows. This macro is for reading a message block from a terminal after a previous READ or WRITE macro has established the line connection.

1. Prepare
2. Read Text
3. Write EOT (TVR only)
4. Disable (TVR only)

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial dials a terminal, writes an EOA to place the terminal in receive state, and writes message text.

1. Disable
2. Dial Dial digits
3. Write Pad characters
4. Write EOA
5. Write Text
6. Write EOT (TIR only)
7. Disable (TIR only)

WRITE Conversational (TV)
WRITE Conversational and Reset (TVR)

WRITE Conversational writes an EOA to place the terminal in receive state and writes message text. This macro is for use following a Read operation to change from receiving text to sending text, when the line connection is already established.

1. Write EOA
2. Write Text
3. Write EOT (TVR only)
4. Disable (TVR only)

WRITE Disconnect (TN)

WRITE Disconnect breaks the line connection.

1. Write EOT
2. Disable

IBM 2740 WITH DIAL-UP AND CHECKING FEATURES

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an answering list, which you define by coding the operand field of a DFTRMLST macro like this:

```
DIALST,0
```

Write Operations

A Write Initial operation requires a calling list, which you define by coding the DFTRMLST operand field like this:

```
DIALST,dialcount,dialchars
```

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)

READ Initial answers a call from a terminal, monitors the line for an EOA sent by the terminal, and reads the message block that follows.

1. Disable
2. Enable
3. Prepare
4. Read Text
5. Write EOA EOT (TIR only)
6. Disable (TIR only)

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same terminal and component that sent the previous block.

1. Write Y
2. Read Text
3. Write EOA EOT (TTR only)
4. Disable (TTR only)

READ Repeat (TP)
READ Repeat and Reset (TPR)

READ Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1. Write N
2. Read Text
3. Write EOA EOT (TPR only)
4. Disable (TPR only)

READ Conversational (TV)
READ Conversational and Reset (TVR)

READ Conversational monitors the line for an EOA sent by the terminal and reads the message block that follows. This macro is for use following a Write operation, to change from sending text to receiving text.

1. Write EOT
2. Prepare
3. Read Text
4. Write EOA EOT (TVR only)
5. Disable (TVR only)

WRITE MACRO INSTRUCTIONS

Programming Note: Each outgoing message block must end with EOB.

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial dials a terminal, writes an EOA to place the terminal in receive state, writes message text, and reads the response to text.

1. Disable
2. Dial Dial digits
3. Write Pad characters
4. Write EOA
5. Write Text
6. Read Response
7. Write EOT (TIR only)
8. Disable (TIR only)

WRITE Continue (TT)
WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1. Write Text
2. Read Response
3. Write EOT (TTR only)
4. Disable (TTR only)

WRITE Conversational (TV)
WRITE Conversational and Reset (TVR)

WRITE Conversational writes an EOA to place the terminal in receive state, writes message text, and reads the response. This macro is for use following a Read operation, to change from receiving text to sending text.

1. Write EOA
2. Write Text
3. Read Response
4. Write EOT (TVR only)
5. Disable (TVR only)

WRITE Continue Conversational (TTV)

WRITE Continue Conversational writes a message block and reads a response from the terminal, then resets it to control mode, monitors the line for an EOA from the terminal and reads the message block that follows.

1. Write Text
2. Read Response
3. Write EOT EOT EOT
4. Prepare
5. Sense
6. Read Text

Programming Note: WRITE TTV performs the same functions as would be performed by a WRITE Continue (TT) followed by a READ Conversational (TV), but saves coding effort by allowing you to verify successful initiation and conclusion of the operation (i.e., by checking return and completion codes) just once, instead of after each of the two separate macro instructions.

WRITE Positive Acknowledgment and Disconnect (TA)

WRITE Positive Acknowledgement and Disconnect writes a positive response to text (an EOA) and breaks the line connection. This macro is for use following a successful READ operation when you wish to break the line connection instead of receiving the remaining blocks of a message.

1. Write EOA EOT
2. Disable

WRITE Negative Acknowledgment and Disconnect (TN)

WRITE Negative Acknowledgment and Disconnect writes a negative acknowledgment (the EOT character serves this purpose) and breaks the line connection. This macro is for use following an unsuccessful Read operation when you wish to break the line connection instead of receiving the remaining blocks of a message. The macro may also be used after a write operation when you wish to break the line connection.

1. Write EOT
2. Disable

IBM 2740 WITH DIAL-UP AND TRANSMIT CONTROL FEATURES

Caution: A 2740 having the Transmit Control feature is equipped with a Transmit Control switch. This switch must always be in the MTC position when the 2740 is under BTAM control.

DEFINING TERMINAL LISTS

A Read Initial or Write Initial operation that answers a call from a terminal requires an answering list, which you define by coding the operand field of a DFTRMLST macro like this:

```
DIALST,0
```

A Read Initial or Write Initial operation that calls a terminal requires a calling list, which you define by coding the DFTRMLST operand field like this:

```
DIALST,dialcount,dialchars
```

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)
(Using Calling List)

READ Initial dials a terminal, writes a selection sequence, and if the response is positive, reads a message block. If the response is negative, the operation is posted complete.

1. Disable
2. Dial Dial digits
3. Write Pad characters
4. Write Selection sequence
5. Read Response
6. Read Text
7. Write EOT (TIR only)
8. Disable (TIR only)

READ Initial (TI)
READ Initial and Reset (TIR)
(Using Answering List)

READ Initial answers a call from a terminal, writes a selection sequence, and if the response is positive, reads a message block. If the response is negative, the operation is posted complete.

1. Disable
2. Enable
3. Write Pad characters
4. Write Selection sequence
5. Read Response
6. Read Text
7. Write EOT (TIR only)
8. Disable (TIR only)

READ Conversational (TV)
READ Conversational and Reset (TVR)

READ Conversational writes a selection sequence, and if the response is positive, reads a message block. If the response is negative, the operation is posted complete. This macro is for use following a Write operation, to change from sending text to receiving text.

1. Write Selection sequence
2. Read Response
3. Read Text
4. Write EOT (TVR only)
5. Disable (TVR only)

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)
WRITE Initial and Reset (TIR)
(Using Calling List)

WRITE Initial dials a terminal, writes an EOA to place the terminal in receive state, and writes message text to the terminal.

1. Disable
2. Dial Dial digits
3. Write Pad characters
4. Write EOA
5. Write Text
6. Write EOT (TIR only)
7. Disable (TIR only)

WRITE Initial (TI)
WRITE Initial and Reset (TIR)
(Using Answering List)

WRITE Initial answers a call from a terminal, writes an EOA to place it in receive state, and writes message text to the terminal.

1. Disable
2. Enable
3. Write Pad characters
4. Write EOA
5. Write Text
6. Write EOT (TIR only)
7. Disable (TIR only)

WRITE Conversational (TV)
WRITE Conversational and Reset (TVR)

WRITE Conversational writes message text to the terminal. This macro is for use following a Read operation, to change from receiving text to sending text.

1. Write EOA
2. Write Text
3. Write EOT (TVR only)
4. Disable (TVR only)

WRITE Disconnect (TN)

WRITE Disconnect breaks the line connection.

1. Write EOT
2. Disable

IBM 2740 WITH DIAL-UP, TRANSMIT CONTROL, AND CHECKING FEATURES

Caution: A 2740 having the Transmit Control feature is equipped with a Transmit Control switch. This switch must always be in the MTC position when the 2740 is under BTAM control.

DEFINING TERMINAL LISTS

A Read Initial or Write Initial operation that answers a call from a terminal requires an answering list, which you define by coding the operand field of a DFTRMLST macro like this:

```
DIALST,0
```

A Read Initial or Write Initial operation that calls a terminal requires a calling list, which you define by coding the DFTRMLST operand field like this:

```
DIALST,dialcount,dialchars
```

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)
(Using Calling List)

READ Initial dials a terminal, writes a selection sequence, and if the response is positive, reads a message block. If the response is negative, the operation is posted complete.

1. Disable
2. Dial Dial digits
3. Write Pad characters
4. Write Selection sequence
5. Read Response
6. Read Text
7. Write EOA EOT (TIR only)
8. Disable (TIR only)

READ Initial (TI)
READ Initial and Reset (TIR)
(Using Answering List)

READ Initial answers a call from a terminal, writes a selection sequence, and if the response is positive, reads a message block.

1. Disable
2. Enable
3. Write Pad characters
4. Write Selection sequence
5. Read Response
6. Read Text
7. Write EOA EOT (TIR only)
8. Disable (TIR only)

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same terminal and component that sent the previous block.

1. Write Y
2. Read Text
3. Write EOA EOT (TTR only)
4. Disable (TTR only)

READ Repeat (TP)
READ Repeat and Reset (TPR)

READ Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1. Write N
2. Read Text
3. Write EOA EOT (TPR only)
4. Disable (TPR only)

READ Conversational (TV)
READ Conversational and Reset (TVR)

READ Conversational writes a selection sequence, and if the response is positive, reads a message block. If the response is negative, the operation is posted complete. This macro is for use following a Write operation, to change from sending text to receiving text.

1. Write EOT EOT EOT
2. Write Selection sequence
3. Read Response
4. Read Text
5. Write EOA EOT (TVR only)
6. Disable (TVR only)

WRITE MACRO INSTRUCTIONS

Programming Note: Each outgoing message block must end with EOB.

WRITE Initial (TI) WRITE Initial and Reset (TIR) (Using Calling List)

WRITE Initial dials a terminal, writes an EOA to place the terminal in receive state, writes message text to the terminal, and reads a response from the terminal.

1. Disable
2. Dial — Dial digits
3. Write — Pad characters
4. Write — EOA
5. Write — Text
6. Read — Response
7. Write — EOT (TIR only)
8. Disable (TIR only)

WRITE Initial (TI) WRITE Initial and Reset (TIR) (Using Answering List)

WRITE Initial answers a call from the terminal, writes an EOA to place it in receive state, writes message text to the terminal, and reads a response from the terminal.

1. Disable
2. Enable
3. Write — Pad characters
4. Write — EOA
5. Write — Text
6. Read — Response
7. Write — EOT (TIR only)
8. Disable (TIR only)

WRITE Continue (TT) WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1. Write — Text
2. Read — Response
3. Write — EOT (TTR only)
4. Disable (TTR only)

WRITE Conversational (TV) WRITE Conversational and Reset (TVR)

WRITE Conversational writes an EOA to place the terminal in receive state, writes message text, and reads the response. This macro is for use following a Read operation, to change from receiving text to sending text.

1. Write — EOA
2. Write — Text
3. Read — Response
4. Write — EOT (TVR only)
5. Disable (TVR only)

WRITE Positive Acknowledgment and Disconnect (TA)

WRITE Positive Acknowledgment and Disconnect writes a positive response to text (an EOA) and breaks the line connection. This macro is for use following a successful Read operation when you wish to break the line connection instead of receiving the remaining blocks of a message.

1. Write — EOA EOT
2. Disable

WRITE Negative Acknowledgment and Disconnect (TN)

WRITE Negative Acknowledgment and Disconnect writes a negative acknowledgment (the EOT character serves this purpose) and breaks the line connection. This macro is for use following an unsuccessful Read operation when you wish to break the line connection instead of receiving the remaining blocks of a message. The macro may also be used after a write operation when you wish to break the line connection.

1. Write — EOT
2. Disable

IBM 2740 WITH STATION CONTROL FEATURE

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an open or wraparound polling list. The list may have one or more terminal entries, each containing a single polling character that identifies the terminal. To define a polling list, code the operand field of a DFTRMLST macro like this:

```
{OPENLST}
{WRAPLST},(xx,...)
```

Write Operations

A Write Initial operation requires an addressing list having a single entry, containing a single addressing character that identifies the terminal that is to receive the output message. To define an addressing list, code the operand field of a DFTRMLST macro like this:

```
OPENLST ,xx
```

READ MACRO INSTRUCTIONS

READ Initial (TI)

READ Initial successively polls the terminals in the polling list, and upon receiving a positive response to polling, reads a message block.

1. Write EOT EOT EOT
2. Write Polling character
3. Write Space character
4. Read Response
5. Read Text

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial addresses a terminal, and if the response is positive, writes an EOA to set the terminal to receive state and writes message text to the terminal. If the terminal sends a negative response to addressing, the operation is posted complete.

1. Write EOT EOT EOT S
2. Write Addressing character
3. Write Space character
4. Read Response
5. Write EOA
6. Write Text
7. Write EOT EOT EOT (TIR only)

IBM 2740 WITH STATION CONTROL AND CHECKING FEATURES

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an open or wraparound polling list. The list may have one or more terminal entries, each containing a single polling character that identifies the terminal. To define a polling list, code the operand field of a DFTRMLST macro like this:

```
{OPENLST}
{WRAPLST},(xx,...)
```

Write Operations

A Write Initial operation requires an addressing list having a single entry, containing a single addressing character that identifies the terminal that is to receive the output message. To define an addressing list, code the operand field of a DFTRMLST macro like this:

```
OPENLST,xx
```

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)

READ Initial successively polls the terminals in the polling list, and upon receiving a positive response to polling, reads a message block.

1. Write EOT EOT EOT
2. Write Polling character
3. Write Space character
4. Read Response
5. Read Text
6. Write EOA EOT EOT EOT (TIR only)

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same terminal and component that sent the previous block.

1. Write Y
2. Read Text
3. Write EOA EOT EOT EOT (TTR only)

READ Repeat (TP)
READ Repeat and Reset (TPR)

READ Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1. Write N
2. Read Text
3. Write EOA EOT EOT EOT (TPR only)

WRITE MACRO INSTRUCTIONS

Programming Note: Each outgoing message block must end with EOB.

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial addresses a terminal, and if the response to addressing is positive, writes EOA followed by message text and reads the response from the terminal.

1. Write EOT EOT EOT S
2. Write Addressing character
3. Write Space character
4. Read Response
5. Write EOA
6. Write Text
7. Read Response
8. Write EOT EOT EOT (TIR only)

WRITE Continue (TT)
WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1. Write Text
2. Read Response
3. Write EOT EOT EOT (TTR only)

WRITE Positive Acknowledgment (TA)

WRITE Positive Acknowledgment writes a positive acknowledgment and an EOT sequence to indicate to the terminal that the computer received message text without error and to stop line activity. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write EOA EOT EOT EOT

WRITE Negative Acknowledgment (TN)

WRITE Negative Acknowledgment writes an EOT sequence to indicate to the terminal that the computer received text with an error and to stop line activity. The terminal interprets the EOT sequence as a negative response. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write EOT EOT EOT

IBM 2760 OPTICAL IMAGE UNIT - GENERAL INFORMATION

OPERATION AND MESSAGE FORMATS

This section describes the essential functions of the IBM 2760 Optical Image Unit and explains their relationship to the formats of messages to and from the computer. (See the 2760 component description publication listed in the Preface for detailed information on this terminal device.)

FILMSTRIP POSITIONING

After the operator inserts the filmstrip cartridge into the front of the 2760, the filmstrip drive mechanism positions the filmstrip in accordance with instructions from the user program. These instructions are received in the form of a message of predefined format, called a frame change message. Three characters, designated F, A_1, and A_2, determine how the filmstrip is to be positioned. The F (function) character specifies the direction of filmstrip movement (this character has other functions, discussed below). The A_1 and A_2 characters are codes indicating the amount (i.e., number of frames) of filmstrip movement. Figure 14 gives the meanings of the possible values of the F character; Figure 15 gives the codes for the A_1 and A_2 characters.

F Char.	Film Movement	Mode of Operation	Manual Frame Advance
Space	Reverse	Auto EOM	Disabled
1	Forward	Auto EOM	Disabled
2	Reverse	Manual EOM	Disabled
3	Forward	Manual EOM	Disabled
4	Reverse	Auto EOM	Enabled
5	Forward	Auto EOM	Enabled
6	Reverse	Manual EOM	Enabled
7	Forward	Manual EOM	Enabled

Figure 14. F (Function) Character Codes

Frames of Film Movement or Number in Image Index Counter	Character Transmitted A_1/I_1	Frames of Film Movement or Number in Image Index Counter	Character Transmitted A_2/I_2
0	Space (C Bit)	0	Space (C Bit)
32	@	1	@
64	—	2	—
96	&	3	&
128	1	4	1
160	/	5	/
192	j	6	j
224	a	7	a
256	2	8	2
288	s	9	s
320	k	10	k
352	b	11	b
384	3	12	3
416	t	13	t
448	ℓ	14	ℓ
480	c	15	c
		16	4
		17	u
		18	m
		19	d
		20	5
		21	v
		22	n
		23	e
		24	6
		25	w
		26	o
		27	f
		28	7
		29	x
		30	p
		31	g

Figure 15. A_1/I_1 and A_2/I_2 Character Codes

MODES OF OPERATION

In addition to indicating the direction of filmstrip movement, the F character designates whether the unit is to operate in Manual or Automatic EOM mode and whether or not in Manual Frame Advance mode.

Manual vs. Automatic EOM Mode

The 2760 operates in one of two modes when sending a message to the computer. In automatic EOM (end of message) mode the message contains a single set of response point coordinates (explained below), and the message is sent automatically when the terminal operator probes a response point.

In manual EOM mode, more than one set of response point coordinates can be sent in the same message. In this mode, the terminal operator indicates the end of the message by probing the End Entry response point (one of the three Utility response points).

Manual Frame Advance

Positioning of the filmstrip is ordinarily performed upon instruction from the user program. The filmstrip may alternatively be positioned by the terminal operator; this, in conjunction with Manual EOM mode, allows the operator to enter a message containing response points from more than one image (frame). Use of this feature results in fewer program interruptions for repositioning the filmstrip and is therefore more economical of CPU time. Its use also simplifies logical program organization in that the program does not have to provide a frame change message for every possible circumstance.

The terminal operator moves the filmstrip by means of the Film switch on the front of the 2760. He can use the switch only when it has been made operational ("enabled") by the program. The F character sent by the program determines whether the switch is enabled or disabled.

RESPONSE POINTS AND COORDINATES

Each of the two halves of the Optical Image Unit screen, the image screen (right half) and auxiliary screen (left half) has 120 possible response points, in a 10 (horizontal) by 12 (vertical) matrix. Only a few, or perhaps one, of these points will be utilized in any given image (frame) or overlay. Each response point is represented by a set of vertical and horizontal (V and H) coordinates. It is these coordinates that are sent to the computer when the operator probes a response point. The user program must contain a table that associates with each valid response point some value or bit setting representing the response probed by the operator.

Figure 16 gives the V and H coordinates for each of the response points. Each coordinate is represented by a character, which on the communication line is represented by the corresponding bit pattern in transmission code.

Horizontal Positions

	1	2	3	4	5	6	7	8	9	10		11	12	13	14	15	16	17	18	19	20	21	
L	l5	ld	lm	lu	l4	lc	ll	lt	l3	lb	1	lk	ls	l2	la	lj	l/	l1	l&	l-	l@		
T	t5	td	tm	tu	t4	tc	tl	tt	t3	tb	2	tk	ts	t2	ta	tj	t/	t1	t&	t-	t@		
3	35	3d	3m	3u	34	3c	3l	3t	33	3b	3	3k	3s	32	3a	3j	3/	31	3&	3-	3@		
B	b5	bd	bm	bu	b4	bc	bl	bt	b3	bb	4	bk	bs	b2	ba	bj	b/	b1	b&	b-	b@		
K	k5	kd	km	ku	k4	kc	kl	kt	k3	kb	5	kk	ks	k2	ka	kj	k/	k1	k&	k-	k@	k space	UNLOAD
S	s5	sd	sm	su	s4	sc	sl	st	s3	sb	6	sk	ss	s2	sa	sj	s/	s1	s&	s-	s@	s space	LOAD
2	25	2d	2m	2u	24	2c	2l	2t	23	2b	7	2k	2s	22	2a	2j	2/	21	2&	2-	2@		
A	a5	ad	am	au	a4	ac	al	at	a3	ab	8	ak	as	a2	aa	aj	a/	a1	a&	a-	a@	a space	END ENTRY
J	j5	jd	jm	ju	j4	jc	jl	jt	j3	jb	9	jk	js	j2	ja	jj	j/	j1	j&	j-	j@		
/	/5	/d	/m	/u	/4	/c	/l	/t	/3	/b	10	/k	/s	/2	/a	/j	//	/1	/&	/-	/@		
1	15	1d	1m	1u	14	1c	1l	1t	13	1b	11	1k	1s	12	1a	1j	1/	11	1&	1-	1@		
&	&5	&d	&m	&u	&4	&c	&l	&t	&3	&b	12	&k	&s	&2	&a	&j	&/	&1	&&	&-	&@		
	5	D	M	U	4	C	L	T	3	B		K	S	2	A	J	/	1	&	-	@ ←	(H) Horizontal Coordinates*	

(V) Vertical Coordinates* Auxiliary Screen Vertical Positions Image Screen

* Shown in upper case for ease of reading. The 2760 operates in lower case shift automatically. No case shift characters are required or permitted in messages to or from the unit.

Figure 16. V & H (Vertical & Horizontal) Response Point Coordinate Codes

Utility Response Points

Three special response points appear in a vertical row to the right of the image screen. The Load response point, when probed, indicates to the user program that the terminal operator has inserted a filmstrip cartridge into the Optical Image Unit. Upon receiving the V and H coordinates of this response point, the user program should send a frame change message to the 2760 that causes it to advance the filmstrip to the first frame.

The Unload response point, when probed, indicates to the user program that it should send a message that retracts the filmstrip into the cartridge and ejects the cartridge.

The End Entry response point is probed at the end of each message, when the 2760 is operating in Manual EOM Mode.

IMAGE INDEX COUNTER

Within the Optical Image Unit is an electronic counter called an Image Index Counter, which is incremented and decremented in step with filmstrip movement. The counter thus maintains a continuous record of which frame of the filmstrip is currently being projected. The content of this counter is transmitted to the computer at the beginning of each message to the computer or, when manual frame advance is being used, at the beginning of each sequence of response points from a given image. The two characters representing the content of the counter are designated I_1 and I_2.

Upon receipt by the computer of each message from the 2760, the user program should check the I_1 I_2 characters to ensure that the correct image is being displayed.

Figure 15 shows the characters that represent the numerical content of the image index counter.

If the program sends to the 2760 a message that instructs the mechanism to move the filmstrip to a point that is beyond its last frame, an interlock is activated that prohibits filmstrip movement. This in turn prevents the image index counter from being incremented.

MESSAGE FORMATS

Each message from the computer to the IBM 2760 begins with the sequence EOA PRE o (end-of-address, prefix, lowercase o). The PRE o characters indicate that the message is intended for the 2760 and not for the IBM 2740 with which it is associated. Similarly, each message from the 2760 to the computer begins with EOA PRE o. The user program should check the input area for the presence of these two characters. (The PRE o sequence is represented by X'3E4C' in transmission code.) Each message in either direction ends with an EOB (end-of-block) character.

Figure 17 presents the formats for messages between the computer and the Optical Image Unit.

Computer to IBM 2760

Output messages, referred to as frame change messages, direct the Optical Image Unit to move the film forward or backward, set the mode for the subsequent response message, and specify the amount of filmstrip travel (i.e., number of frames). See Figure 14 for the meanings of the possible values of the F character; see Figure 15 for the coded values representing amount of filmstrip movement (A_1 and A_2 characters).

IBM 2760 to Computer

Input messages, referred to as response messages, indicate to the user program which filmstrip frame is being displayed for the current response, and give the V and H coordinates of the response point or points the operator has probed.

Auto EOM Mode: In this mode, each probe action by the terminal operator causes a complete message, containing the coordinates of one response point, to be sent to the computer.

Manual EOM Mode, Film Switch Disabled: In this mode, a message may contain any number of response point coordinates. The sequence EOA PRE o I_1 I_2 V_1 H_1 is sent to the computer when the first response is probed. Each subsequent set of V and H coordinates is sent individually as each response point is probed. The EOB character is automatically sent after the coordinates when the End Entry response point is probed.

The user program should check for the End Entry response point to ensure that the message contains the correct number of responses.

Manual EOM Mode, Film Switch Enabled: In this mode, the terminal operator may reposition the filmstrip to a new frame while entering response points, so that the input message contains responses from more than one image. The image index counter is incremented or decremented each time the filmstrip is moved.

The first probe action by the terminal operator following manual positioning of the filmstrip causes the sequence PRE o I_1 I_2 V_1 H_1 ... to be sent to the computer. Figure 17 illustrates the message resulting from entering response points for three different images.

The only times the operator cannot move the filmstrip, when the Film Switch is enabled, are when the image index counter contains a value of 2 or less, in which case reverse movement is inhibited; when the 2760 detects the hole in the tenth trailer frame, in which case forward movement is inhibited; and when the 2760 is at that moment receiving or executing a message from the computer.

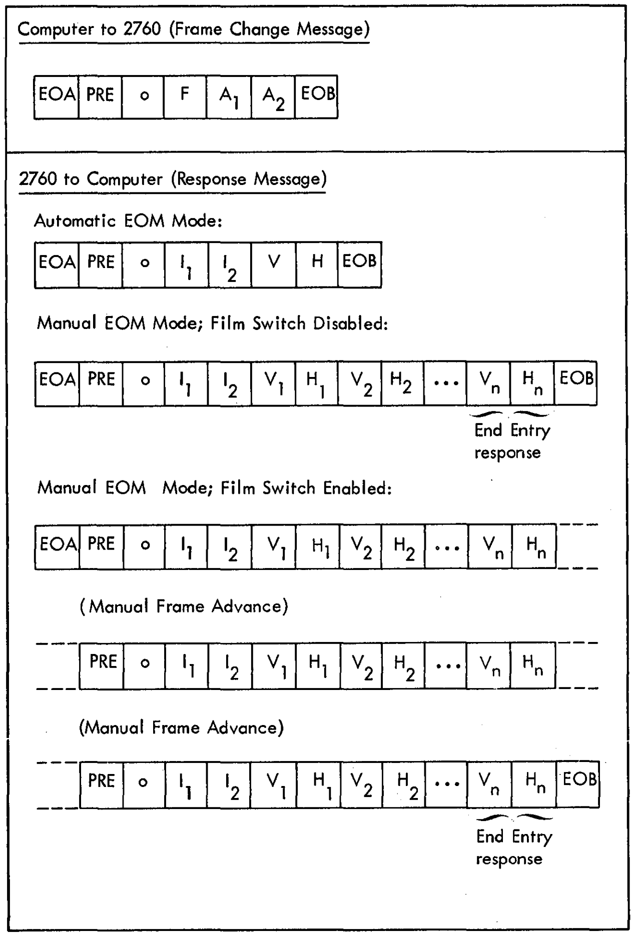


Figure 17. Message Formats

SIGNALS TO OPERATOR

Two kinds of signals inform the terminal operator that his probe actions are correct. One is a visible indicator: the On-Line light. The second is an audible tone. The use of these signals differs for Automatic EOM, and Manual EOM mode.

Automatic EOM Mode

When the operator enters responses in this mode, the On-Line light comes on when he probes the response point. The audible tone sounds and the light goes out when the computer has returned a positive answer-back, indicating that it received the message without error. The positive answer-back may be a Y or an EOA (see Error Detection and Recovery).

In describing 2760 operations, the term answerback is used instead of response, to avoid confusion with responses entered by the terminal operator.)

Manual EOM Mode

In this mode, the On-Line light comes on when the terminal operator probes the first response point and remains on throughout the remainder of the message. This should alert the operator that the terminal is in Manual EOM mode and accordingly is subject to the inter-character time-out imposed by the transmission control unit. That is, each subsequent response should be made within that time limit[1]. The On-Line light goes off when the computer replies with a Y, EOA, or EOT.

The audible tone sounds as each response point is probed, to inform the operator that the response has been sent to the computer and that he may probe again. The final probe action (End Entry) results in the tone only after the computer replies with a Y or EOA. This signifies to the operator that the complete message was received without error.

If a response point is incorrectly probed, the tone does not sound and the Probe Check light appears. This indicates that the probe touched the screen at too small an angle from the vertical or that it touched outside the designated response point area.

ERROR DETECTION AND RECOVERY

Messages to and from the IBM 2760 are checked for errors by the Record Checking facility of the 2740 to which the 2760 is attached. In addition, the 2760 checks messages from the computer for proper length; all messages from the computer have the same length.

If a text error or record length error is detected in a frame change message, the 2760/2740 sends a N (negative answerback) character in response to the EOB that ends the frame change message. If the EROPT operand of the DCB macro for the line group specifies W (retry of write-text errors),

---

[1]It may be well to specifically inform the operator of the time limit by means of a suitable phrase on the image, for each frame for which the limit applies.

BTAM error recovery procedures cause the frame change message (without the EOA character) to be retransmitted up to two more times. If the error condition persists beyond the three attempts, or if EROPT does not specify W, the error condition is posted in the data event control block (DECB) for the line.

If the frame change message is received without error, the 2760/2740 sends a Y (positive answerback) character. BTAM responds to the Y with an EOT, which causes the 2760 to execute the instructions contained in the frame change message.

If a text error is detected in a response message, and the EROPT operand of the DCB macro for the line group specifies R (retry of read-text errors), BTAM error recovery procedures send a N to the 2760/2740, then reread the response message. If the Optical Image Unit is in Automatic EOM mode, it automatically resends the message; if it is in Manual EOM mode, the terminal operator must re-enter the entire response message.

The error recovery procedures respond with N and reread the message up to two more times. If the error condition persists beyond the three attempts, or if EROPT does not specify R, the error condition is posted in the DECB for the line.

If the response message is received without error, the operation on which the message was received is posted complete. The EOA character that begins the next frame change message serves as a positive answerback to the 2760/2740.

It is possible that the terminal operator will probe the screen of the 2760 at a time when the probe is activated but there is no Read command in effect to receive the data. Should this occur, the next operation executed for the line to which that 2760 is connected will be posted complete-with-error, indicating that probe data was lost. In order to recover from an error of this kind, you may wish to take one of these suggested actions:

- Issue a WRITE TCO macro that moves the filmstrip to an error-handling frame that will aid the terminal operator in recovering the lost data.
- Issue a WRITE TV macro to write an error message on the printer of the 2740 to which the 2760 is attached.
- Issue a READ TI macro (nonswitched line) or READ TV macro (switched line) to read the next message from the terminal. The operator should be instructed to re-probe the previous response if he does not hear the audible tone within a reasonable interval after probing.

For information on error indicators at the Optical Image Unit, see the <u>IBM 2760 Optical Image Unit - Component Description</u> publication.

ON-LINE TESTING

On-line tests for the IBM 2760 are initiated at the IBM 2740 terminal keyboard. The test request message can be keyed in whenever the user program issues a READ macro instruction (other than READ Skip) without the Reset option, or it may be keyed in after the filmstrip has been moved by a WRITE TCO macro. In order to use the on-line test facility, the EROPT field in the DCB for the line group must specify T.

Two tests are available for the 2760: frame change test (type 10) and scan point test (type 11). They are designed to test the filmstrip transport mechanism and the probe response accuracy of the 2760. See Test Type Codes in the chapter On-Line Testing for descriptions of these tests.

On-line tests will generally be run by the IBM Customer Engineer during periods of inactivity or as a startup procedure. Once the testing is completed, the Customer Engineer will unload the filmstrip and the operator can continue operation.

The terminal operator can also interrupt his data entry operation at any time to run a terminal test. However, some provision must be made to reposition the filmstrip to the frame being displayed when the test was begun, and to reset the modes in effect at the time. A recommended way of doing this is for the user program to save (1) the image index value expected in the response message that would have been received had not the test request message been received instead, and (2) the F-character contained in the last previous frame change message.

Then, upon conclusion of the test, the program would (1) calculate the difference between the values of the image index at the beginning and at the end of the test, and from this difference determine the A_1 and A_2 characters to be sent in the next frame change message; and (2) determine, from the modes at the beginning of the test and the sign of the difference in the image index values, the appropriate F-character for the frame change message. Sending that message to the 2760 would properly position the filmstrip and set the correct modes.

As an example, assume that the filmstrip was positioned at frame 27 at the beginning of the test and that the modes in effect were Manual EOM, Manual Frame Advance disabled. If the previous filmstrip movement had been in the forward direction, the last F-character sent would be 3 (see Figure 14). Then assume that at the end of the test the filmstrip was positioned at frame 44. To reposition it to frame 27 would require a reverse movement of 17, hence the A_1, A_2 characters in the next frame change message would be Space U (see Figure 15) and the F-character would specify reverse direction. To restore the modes to their original settings (Manual EOM, Manual Frame Advance disabled), the F-character should be 2. In computing the F-character, it is useful to know that the filmstrip direction is determined by bit 6 (0=reverse, 1= forward), the Manual Frame Advance mode is determined by bit 4 (0=disabled, 1= enabled), and the EOM mode is determined by bit 5 (0=automatic, 1=manual).

To request one of the 2760 tests, the following message must be keyed from the 2740:

9 9 9 9 9 x x 4 F A_1 A_2 EOT

xx	=	10 for frame change test
	=	11 for scan point test
4		specifies that the test is for the 2740 terminal (of which the 2760 is a component)
F	=	function control character
A_1 A_2	=	amount of film movement

If the operation in effect when a test request message is entered is a Write Invitational Optical (TCO) operation, the online test facility performs, at the conclusion of the test, a Read Initial (TI) operation (nonswitched line) or Read Conversational (TV) operation, to receive the next regular message block.

PROGRAMMING CONSIDERATIONS

General Steps for Preparing the User Program

1. Define the information the user program is to obtain from the terminal operator.

2. Divide the required information into questions and statements that are meaningful to the terminal operators and that can be answered by probing response points on the Optical Image Unit screen. Responses may be Yes-No choices, multiple choices, alphabetic or numeric data, etc.

3. Design the sequence of questions to request the information in the most efficient order. Make sure that the questions are coordinated so that each piece of information is obtained at the proper point in the data entry procedure.

4. Make a preliminary design of all the filmstrip frames that request information. Decide on the wording of the questions and the wording and location of the response points.

5. Design, on the basis of the frame layouts, a system of tables that will enable the user program to recognize the valid and invalid responses to a question on a frame.

6. Make a final design of all the frames that request information, and modify the tables as necessary to make them more efficient.

7. Arrange the filmstrip layout to minimize film movement (e.g., error frames and other frequently displayed frames should be in the middle of the filmstrip).

8. Design a method of initialization so that the program can locate the frame containing the first application image that requests information from the operator even if some of the leader frames have been cut off. The section, Initializing Images, suggests a means for doing this.

9. Design the frames necessary for initialization.

10. Code the user program on the basis of the tables and frames. The program must include the initialization routine as well as the error routine for detecting and notifying the operator of invalid responses.

Initializing Images

Once the filmstrip cartridge is inserted in the Optical Image Unit, five steps of the filmstrip drive mechanism are required to advance the filmstrip out of the cartridge to the point where the first frame can be displayed. It is recommended that three blank frames be left at the beginning of the filmstrip to serve as a leader, to absorb the greater wear that the beginning of the strip receives. The first frame beyond the blank frames is called the initializing image. This image contains a probe response point whose position is unique for that filmstrip, thus serving to identify the filmstrip to the program so that the correct data entry procedure routine can be determined.

A further recommendation is that several initializing images be used, so that a new leading edge can be trimmed on the filmstrip as the original edge becomes worn through use. Each of the initializing images would have response points whose meanings are the same, but whose position differs with each frame. Thus the program, when receiving a response message following the first frame change message, can identify which of the initializing images is being displayed. This information can then be used to modify the Image Index values received in subsequent response messages, thereby compensating for any change in position of the application images (relative to the leading edge of the filmstrip) caused by removing worn frames.

Assume, for example, that a new filmstrip has three blank frames, followed by three initializing frames. The first frame change message would specify a film movement of eight frames. When the filmstrip is new, this message causes the first initializing image to be projected. When the first blank frame is cut off, to provide a new leading edge, the same frame change message will cause the second initializing image to be displayed. Since all subsequent application images are now one frame closer to the leading edge of the filmstrip, it is necessary to subtract one from each image index value received in response messages, in order for the program to correctly identify the frame being displayed for the response points received. By determining which initializing image is being displayed, the program can set the proper decrement value in an index register to modify the received image index values.

Startup Procedure

When the terminal operator is ready to begin a data entry operation, he inserts the appropriate filmstrip cartridge into the Optical Image Unit and probes the Load response point. (If the terminal is connected to the computer by a switched line, he must dial the computer before probing the Load response point.) For either a switched or a nonswitched line, the Standby light on the 2740 must be on and a Read Initial operation must be in progress at the computer. When the operator probes the Load response point, a message containing the coordinates of that point is transmitted to the computer. The program should check the input area for the presence of the Load coordinates, s Space (X'2501'), and upon detecting them, issue a WRITE macro that sends the appropriate frame change message to the 2760. The A_1 and A_2 characters in the frame change message should specify sufficient frame movement to cause the initializing image to be displayed. Assuming that the initializing image is the fourth frame from the beginning of the filmstrip, a forward movement of eight frames is required to position that image for projection.

When the operator probes the response point for the initializing image, the program should issue a frame change message that positions the filmstrip to the first application image to be viewed by the operator. If more than one initializing image is used, as recommended under Initializing Images, the particular frame being viewed by the operator when he enters his response determines the value of the A_1, A_2 characters in the frame change message. For example, assume that the first application image to be viewed is ten frames from the first initializing image. A response entered for the first initializing image should cause the next frame change message to specify eight frames of filmstrip movement. If the response was entered for the second initializing image, however, the frame change message would have to specify a filmstrip movement of seven frames to reach the first application image.

The values of A_1, A_2 in all subsequent frame change messages would be independent of which initializing image was displayed.

Receiving Multiple Message Blocks

Following receipt of a message block from a 2760, the computer must reply with a positive answerback. In the usual conversational operation, in which a sequence of WRITE TCO macros alternates the sending of frame change messages and the receiving of response messages, the macro itself supplies as the positive answerback the EOA character that begins the next frame change message.

If, however, instead of alternating messages in this manner you wish to receive a message from the 2760 and then receive a message block from the 2740 or the 2760, you should use the READ Continue with Leading Acknowledgment (TTA) macro. This macro sends a positive acknowledgment to the 2760 and then receives another message block from the terminal.

The recommended method for receiving multiple probes from the same filmstrip frame is to specify, in the frame change message that positions that frame, a function character specifying Manual EOM mode. If more than one message block is required to accommodate the response data to be entered from that frame, you should specify Manual EOM mode and receive subsequent response messages by means of a READ TTA macro or a WRITE TCO macro that specifies no filmstrip movement.

You should not issue a READ Continue (TT) macro following receipt of a response message from the 2760, for the following reason. READ TT sends a Y (positive answerback) to the 2760, which causes the 2760 to return an EOT to the computer and to release the probe interlock, which allows the terminal operator to probe response points. When the Read Continue operation receives the EOT, the operation is posted complete; therefore, no Read operation is in effect to receive the next message from the terminal. (Both Y and EOA are recognized by the terminal as a positive answerback; however, the Y causes the terminal to reply with an EOT but the EOA does not.)

Sending Message Blocks Alternately to the 2760 and 2740

If it is necessary to send message blocks alternately to the 2760 and the 2740, the message block to the 2760 should be sent first, followed by the message block to the 2740. If the line is not put in control mode after sending to the 2760, the message block to the 2740 must begin with text and must end with an EOB. If the line is put in control mode after sending to the 2760, the message block to the 2740 must begin with an EOA and must end with an EOB.

IBM 2740 WITH CHECKING FEATURE AND 2760 OPTICAL IMAGE UNIT

Read and Write operations for the 2740 with the checking feature and 2760 Optical Image Unit require no terminal lists.

Once the line group data set has been opened, a READ Initial macro may be issued to establish contact with the 2740 to which the 2760 is attached. If the terminal operator wishes to send from the 2740 keyboard, he presses the Bid key and enters the data. If he wishes to begin data entry with the Optical Image Unit, he inserts the appropriate cartridge into the front of the unit and touches the probe to the Load response point. Either action causes the data to be placed in main storage at the location specified by the area operand of the READ macro.

The user program can determine whether the message came from the keyboard or the Optical Image Unit by testing for the presence of the PRE o characters at the beginning of the input area.

All data sent to the Optical Image Unit is in the form of the fixed-length message EOA PRE o F A_1 A_2 EOB, where F represents the function control character and A_1 and A_2 are characters designating the amount of filmstrip movement as explained under IBM 2760 Optical Image Unit -- General Information.

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)

READ Initial monitors the line for an EOA sent by the terminal and reads the message block that follows.

1.	Write	EOT EOT EOT
2.	Prepare	
3.	Sense	
4.	Read	Text
5.	Write	EOA EOT EOT EOT (TIR only)

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same terminal and component that sent the previous block.

1.	Write	Y
2.	Read	Text
3.	Write	EOA EOT EOT EOT (TTR only)

READ Continue with Leading Acknowledgment (TTA)

READ Continue with Leading Acknowledgment is for use when you wish to positively acknowledge a message, reset the terminal to standby status, and receive message text from either the 2760 or the 2740.

READ TTA should be used specifically to receive message text from the 2740 or 2760 following receipt of text from the 2760. The EOA is a positive acknowledgment (answerback) to the message block received from the 2760, and causes the audible tone to sound and the 2760 probe to become activated (i.e., releases the probe interlock). The EOT sequence resets the terminal to standby status so that either the 2760 or the 2740 may enter message text.

1.	Write	EOA EOT EOT EOT
2.	Prepare	
3.	Sense	
4.	Read	Text

READ Repeat (TP)
READ Repeat and Reset (TPR)

READ Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1.	Write	N
2.	Read	Text
3.	Write	EOA EOT EOT EOT (TPR only)

WRITE MACRO INSTRUCTIONS

Programming Note: Each outgoing message block must end with EOB.

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial writes an EOA to place the terminal in receive state and turn on the terminal motors, writes message text, and reads the response.

1.	Write	EOA and 15 Idle characters

2. Write Text
3. Read Response
4. Write EOT EOT EOT (TIR only)

WRITE Continue (TT)
WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1. Write Text
2. Read Response
3. Write EOT EOT EOT (TTR only)

WRITE Conversational (TV)
WRITE Conversational and Reset (TVR)

WRITE Conversational writes an EOA to place the terminal in receive state, writes message text, and reads the response. This macro is for use following a Read operation, to change from receiving text to sending text.

1. Write EOA
2. Write Text
3. Read Response
4. Write EOT EOT EOT (TVR only)

WRITE Initial Optical (TIO)

This option is for use when you wish to send a frame change message, but do not require a response from the terminal operator (as when retracting the filmstrip and ejecting the cartridge at the end of a data entry operation), or when you wish to receive the response using a subsequent macro. The macro writes the sequence EOA PRE o, to indicate to the 2740 terminal that the message is intended for the 2760, writes the frame change characters and the EOB character, then reads the answerback (response to checking).

If the answerback is positive, the macro ends the operation by sending an EOT to the terminal. If the answerback is negative, the channel program is ended at this point and the error condition is posted in the DECB for the line, except that if Write retries are specified (EROPT=W in DCB), BTAM error recovery procedures resend the frame change characters up to two additional times before posting the error condition.

You must specify in the entry operand of the WRITE TIO macro the address of the main storage location containing the three-character F A_1 A_2 sequence.

1. Write EOA PRE o
2. Write Frame Change Characters
3. Write EOB
4. Read Answerback
5. Write EOT EOT EOT

Programming Note: If input from the 2760 is expected following execution of the Write Initial Optical operation, you should issue a READ Initial (TI) macro immediately after completion of the Write TIO operation.

WRITE Invitational Optical (TCO)

This option is for use when you wish to send a frame change message and read a response message from the terminal. The macro functions identically to the WRITE Initial Optical (TIO) macro, but in addition receives message text from the Optical Image Unit or the 2740 keyboard. The Prepare command (see below) monitors the line for an EOA character; when it is received, the Read Text command reads into the input area that follows the EOA.

You must specify in the entry operand of the WRITE TCO macro the address of the main storage location containing the F A_1 A_2 sequence; in the area operand you must specify the address of the input area into which the response message is to be received. If dynamic buffering is used to read the response message, you should specify the length operand as 'S'.

The WRITE TCO macro is the principal macro used in a 2760 application, as it is a convenient means for alternately sending frame change messages and receiving responses from the operator.

1. Write EOA PRE o
2. Write Frame Change Characters
3. Write EOB
4. Read Answerback
5. Write EOT EOT EOT
6. Prepare
7. Sense
8. Read Text

Examples of WRITE TIO and WRITE TCO

Figure 18 illustrates how WRITE TIO and WRITE TCO are coded. The WRITE TIO macro sends a frame change message to move the film forward six frames. The WRITE TCO macro sends a frame change message to move the filmstrip forward 37 frames and then

```
          WRITE  DECBNAME,TIO,DCBNAME,,,FRMSG1,,MF=E
            .
            .
            .
          WRITE  DECBNAME,TCO,DCBNAME,INAREA,20,FRMSG2,,MF=E
            .
            .
            .
FRMSG1      DC   X'020143   (1 SP J (LOWERCASE))
FRMSG1      DC   X'02200B'   (1 @ /)
INAREA      DS   5F
```

Figure 18. Examples of WRITE TIO and WRITE TCO Macro Instructions (Nonswitched Line)

reads a response message from the terminal. (The A_1 character, @, represents a film movement of 32 frames (see Figure 15); its hexadecimal equivalent is X'20'. The A_2 character, /, represents a movement of five frames; its hexadecimal equivalent is X'0B'. Together, the two characters specify a film movement of 37 frames.)

In each case, the F character, 1 (X'02'), specifies (in addition to forward movement) that the response from the 2760 is to be made in Automatic EOM mode with the Film switch (Manual Frame Advance) disabled.

WRITE Positive Acknowledgment (TA)

WRITE Positive Acknowledgment writes a positive acknowledgment and an EOT sequence to indicate to the terminal that the computer received message text without error and to stop line activity. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

1. Write EOA EOT EOT EOT

WRITE Negative Acknowledgment (TN)

WRITE Negative Acknowledgment writes an EOT sequence to indicate to the terminal that the computer received text with an error and to stop line activity. The terminal interprets the EOT sequence as a negative response. This macro is for use after a Read operation, when you wish to stop receiving from the terminal before the terminal has sent an EOT.

This macro is also used to cause the 2760 to execute the instructions it received in the preceding frame change message.

1. Write EOT EOT EOT

IBM 2740 WITH DIAL-UP AND CHECKING FEATURES AND 2760 OPTICAL IMAGE UNIT

Once the line group data set has been opened, either the terminal or the computer may establish the line connection. If the terminal is to establish the connection (i.e., dial the computer), issue a READ Initial macro instruction that refers to an answering list. When the terminal operator is ready to enter data, he dials the telephone number of the computer.

If he wishes to send from the 2740 keyboard, he presses the Bid key and enters his data. If he wishes to begin data entry with the Optical Image Unit, he inserts the appropriate cartridge into the front of the unit and touches the probe to the Load response point. Either action causes the data to be placed in main storage at the location specified by the area operand of the READ macro.

The user program can determine whether the message came from the keyboard or the Optical Image Unit by testing for the presence of the PRE o characters at the beginning of the input area.

If the computer is to establish the switched line connection, the WRITE TIO macro, explained below, may be used if you wish to send a frame change message immediately following establishment of the line connection. (Alternatively, the connection can be made using a WRITE Initial macro, with the frame change message being sent by a subsequent WRITE TVO or WRITE TCO macro.)

All data sent to the Optical Image Unit is in the form of the fixed-length message EOA PRE o F A_1 A_2 EOB, where F represents the function control character and A_1 and A_2 are characters designating the amount of filmstrip movement, as explained under IBM 2760 Optical Image Unit--General Information.

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an answering list, which you define by coding the operand field of a DFTRMLST macro like this:

```
DIALST,0
```

Write Operations

A Write Initial operation requires a calling list, which you define by coding the DFTRMLST operand field like this:

```
DIALST,dialcount,dialchars
```

(See WRITE Initial Optical macro for calling list required for that macro.)

READ MACRO INSTRUCTIONS

READ Initial (TI)
READ Initial and Reset (TIR)

READ Initial answers a call from a terminal, monitors the line for an EOA sent by the terminal, and reads the message block that follows.

1.	Disable	
2.	Enable	
3.	Prepare	
4.	Sense	
5.	Read	Text
6.	Write	EOA EOT EOT EOT (TIR only)
7.	Disable	(TIR only)

READ Continue (TT)
READ Continue and Reset (TTR)

READ Continue writes a positive response and reads a message block. This macro is for use following a successful READ Initial (TI), READ Repeat (TP), or another READ Continue to receive another message block from the same terminal and component that sent the previous block.

1.	Write	Y
2.	Read	Text
3.	Write	EOA EOT EOT EOT (TTR only)
4.	Disable	(TTR only)

READ Continue with Leading Acknowledgment (TTA)

READ Continue with Leading Acknowledgment is for use when you wish to positively acknowledge a message, reset the terminal to standby status, and receive message text from either the 2760 or the 2740.

READ TTA should be used specifically to receive message text from the 2740 or 2760 following receipt of text from the 2760. The EOA is a positive acknowledgment (answerback) to the message block received from the 2760, and causes the audible tone to sound and the 2760 probe to become activated (i.e., releases the probe interlock). The EOT sequence resets the terminal to standby status so that either the 2760 or the 2740 may enter message text.

1. Write EOA EOT EOT EOT
2. Prepare
3. Sense
4. Read Text

READ Repeat (TP)
READ Repeat and Reset (TPR)

READ Repeat writes a negative response and reads a message block. This macro is for use following an unsuccessful READ Initial (TI), READ Continue (TT), or another READ Repeat, to receive the same message block read by the previous operation.

1. Write N
2. Read Text
3. Write EOA EOT EOT EOT (TPR only)
4. Disable (TPR only)

READ Conversational (TV)
READ Conversational and Reset (TVR)

READ Conversational monitors the line for an EOA sent by the terminal and reads the message block that follows. This macro is for use following a Write operation, to change from sending text to receiving text.

1. Write EOT EOT EOT
2. Prepare
3. Sense
4. Read Text
5. Write EOA EOT EOT EOT (TVR only)
6. Disable (TVR only)

WRITE MACRO INSTRUCTIONS

Programming Note: Each outgoing message block must end with EOB.

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial dials a terminal, writes an EOA to place the terminal in receive state, writes message text, and reads the response to text.

1. Disable
2. Dial Dial digits
3. Write Pad characters
4. Write EOA
5. Write Text
6. Read Response
7. Write EOT EOT EOT (TIR only)
8. Disable (TIR only)

WRITE Continue (TT)
WRITE Continue and Reset (TTR)

WRITE Continue writes a message block and reads a response from the terminal. This macro is for use following a WRITE Initial (TI) or another WRITE Continue.

1. Write Text
2. Read Response
3. Write EOT EOT EOT (TTR only)
4. Disable (TTR only)

WRITE Conversational (TV)
WRITE Conversational and Reset (TVR)

WRITE Conversational writes an EOA to place the terminal in receive state, writes message text, and reads the response. This macro is for use following a Read operation, to change from receiving text to sending text.

1. Write EOA
2. Write Text
3. Read Response
4. Write EOT EOT EOT (TVR only)
5. Disable (TVR only)

WRITE Initial Optical (TIO)

This option is for use when you wish to establish the line connection and send a frame change message to the Optical Image Unit.

The macro issues a Disable command to disable the line in case this was not done previously, dials the terminal, and writes pad characters to provide time fill to allow the terminal motors to reach operating speed. The macro then writes the sequence EOA PRE o, to indicate to the 2740 terminal that the message is intended for the 2760, writes the frame change characters (F, A_1 and A_2) and the EOB character, then reads the answerback (response to checking).

If the answerback is positive, the macro ends the operation by sending an EOT to the terminal. If the answerback is negative, the channel program is ended at this point and the error condition is posted in the DECB for the line, except that if Write retries are specified (EROPT=W in DCB), BTAM error recovery procedures resend the frame change characters up to two additional times before posting the error condition.

You must specify in the entry operand of the WRITE TIO macro the address of a terminal list defined by a DFTRMLST macro as follows:

```
LIST  DFTRMLST  DIALST,dialcount,
                dialchars,faaseq
```

DIALST specifies the type of list; dialcount and dialchars specify the number of digits in the telephone number and the digits themselves; and faaseq specifies the three characters constituting the frame change message text. faaseq must be coded as the hexadecimal equivalent of the transmission code bit pattern for the desired characters.

```
1.  Disable
2.  Dial     Dial digits
3.  Write    Pad characters
4.  Write    EOA PRE o
5.  Write    Frame change characters
6.  Write    EOB
7.  Read     Answerback
8.  Write    EOT EOT EOT
```

Programming Note: If input from the 2760 is expected following execution of the Write Initial Optical operation, you should issue a READ Conversational (TV) macro immediately after completion of the Write TIO operation.

WRITE Conversational Optical (TVO)

This option is for use when you wish to send a frame change message after the switched line connection has been established, but do not require a response from the terminal operator (as when retracting the filmstrip and ejecting the cartridge at the end of a data entry operation), or when you wish to receive the response using a subsequent macro. The macro writes the sequence EOA PRE o, to indicate to the 2740 terminal that the message is intended for the 2760, writes the frame change characters and the EOB character, then reads the answerback (response to checking).

If the answerback is positive, the macro ends the operation by sending an EOT to the terminal. If the answerback is negative, the channel program is ended at this point and the error condition is posted in the DECB for the line, except that if Write retries are specified (EROPT=W in DCB), BTAM error recovery procedures resend the frame change characters up to two additional times before posting the error condition.

You must specify in the entry operand of the WRITE TVO macro the address of the main storage location containing the three-character F A_1 A_2 sequence.

```
1.  Write    EOA PRE o
2.  Write    Frame Change Characters
3.  Write    EOB
4.  Read     Answerback
5.  Write    EOT EOT EOT
```

Programming Note: If input from the 2760 is expected following execution of the Write Conversational Optical operation, you should issue a READ Conversational (TV) macro immediately after completion of the Write TVO operation.

WRITE Invitational Optical (TCO)

This option is for use after the line connection has been established, when you wish to send a frame change message and read a response message from the terminal. The macro functions identically to the WRITE Conversational Optical (TVO) macro, but in addition receives message text from the Optical Image Unit or the 2740 keyboard. The Prepare command (see below) monitors the line for an EOA character; when it is received, the Read Text command reads into the input area the data that follows the EOA.

You must specify in the entry operand of the WRITE TCO macro the address of the main storage location containing the F A_1 A_2 sequence; in the area operand you must specify the address of the input area into which the response message is to be received. If dynamic buffering is used to read the response message, you should specify the length operand as 'S'.

The WRITE TCO macro is the principal macro used in a 2760 application, as it is a convenient means for alternately sending frame change messages and receiving responses from the operator.

```
1.  Write    EOA PRE o
2.  Write    Frame Change Characters
3.  Write    EOB
4.  Read     Answerback
5.  Write    EOT EOT EOT
6.  Prepare
7.  Sense
8.  Read     Text
```

Examples of WRITE TIO, WRITE TVO, and WRITE TCO

Figure 19 illustrates how WRITE TIO, WRITE TVO, and WRITE TCO macros are coded. The WRITE TIO macro dials the telephone number of the 2740 terminal and sends a frame change message; the F A_1 A_2 sequence is coded in the DFTRMLST macro. In this example, the F character is a Space (X'01'), designating reverse movement, and A_1 and A_2 are both "c" (X'67'), representing a filmstrip movement exceeding the length of the filmstrip. This message therefore causes the filmstrip to be retracted and the cartridge ejected. The A_1 and A_2 characters could alternatively be coded as Space Space (X'0101'), representing zero filmstrip movement, then a subsequent WRITE TVO macro could be used to specify the filmstrip movement. This is useful where the amount of film movement may vary from one loading of the filmstrip to another, and so cannot be specified in a terminal list. The same WRITE TIO would be issued regardless of the film movement needed;the subsequent WRITE TVO would use register notation for the entry operand to provide the needed frame change characters.

The WRITE TVO macro sends a frame change message to move the film forward three frames. The F character, 1 (X'02') specifies Automatic EOM Mode with Film switch (Manual Frame Advance) disabled. The response would be read by a different macro (a READ macro, for example), as the WRITE TVO macro does not read response messages.

The WRITE TCO macro sends a frame change message to move the filmstrip backwards 32 frames, sets the 2760 in Manual EOM mode with the Film switch (Manual Frame Advance) enabled, and reads a response message from the 2760 (or the 2740).

WRITE Positive Acknowledgment and Disconnect (TA)

WRITE Positive Acknowledgment and Disconnect writes a positive response to text (an EOA) and breaks the line connection. This macro is for use following a successful READ operation when you wish to break the line connection instead of receiving the remaining blocks of a message.

1. Write EOA EOT EOT EOT
2. Disable

WRITE Negative Acknowledgment and Disconnect (TN)

WRITE Negative Acknowledgment and Disconnect writes a negative acknowledgment (the EOT character serves this purpose) and breaks the line connection. This macro is for use following an unsuccessful Read operation when you wish to break the line connection instead of receiving the remaining blocks of a message. The macro may also be used after a Write operation when you wish to break the line connection.

1. Write EOT EOT EOT
2. Disable

```
          WRITE     DECBNAME,TIO,DCBNAME,,,LIST,,MF=E
            .
            .
            .
          WRITE     DECBNAME,TVO,DCBNAME,,,FRMSG1,,MF=E
            .
            .
            .
          WRITE     DECBNAME,TCO,DCBNAME,INAREA,20,FRMSG2,,MF=E
            .
            .
            .
LIST      DFTRMLST  DIALST,4,5003,016767     (SP C C (LOWERCASE))
FRMSG1    DC        X'020161'                (1 SP &)
FRMSG2    DC        X'0D2001'                (6 @ SP)
INAREA    DS        5F
```

Figure 19. Examples of WRITE TIO, WRITE TVO, and WRITE TCO Macro Instructions (Switched Line)

IBM 2741 COMMUNICATIONS TERMINAL

GENERAL INFORMATION

The line control scheme for the IBM 2741 differs from that for some other start-stop terminals (e.g., the IBM 1050), in that the terminal and line do not alternate between control mode and text mode, and the polling and addressing functions are absent. Instead the 2741, when in communicate mode (all subsequent discussion presupposes this), alternates between two states: receive and transmit. The 2741 is in a third state, control-receive, between the time it sends an EOT and the time it receives an EOA or EOT from the computer; this state is also entered momentarily when the terminal power switch is turned on or when the mode switch is switched from local to communicate mode. In transmit state, the keyboard is unlocked and the terminal operator can key in data for transmission to the computer. In receive state the keyboard is locked and the terminal can only accept and print data received from the computer. The principal indicator of the state of the terminal is the keyboard. If it is unlocked, allowing the operator to enter data, the terminal is in transmit state; otherwise, it is in receive state.

The terminal alternates between states whenever an end-of-transmission (EOT) character is sent on the line. An EOT sent by the computer always places the terminal in transmit state, and an EOT sent by the terminal always places the terminal in receive state.

The terminal sends an EOT whenever the operator presses the Attention key or the Carrier Return key. These two keys accordingly are the means by which the terminal operator tells the computer that he has finished entering a line of data. The computer sends an EOT whenever it executes a channel program that is to receive data from the terminal, i.e., any Read channel program.

Communication between terminal and computer is always initiated by the terminal operator, and can occur anytime after the program sets up the first Read operation, which must be a Read Initial. Conversation begins when the terminal operator sets the mode switch to Communicate (this action sends an end-of-address (EOA) character to the computer). For switched lines, the operator follows this by manually dialing the telephone number of the computer and switching the common-carrier data set to data mode*.

Line control discipline for the 2741 differs from that for other start-stop terminals in the following significant respect. With most terminals, control of the communication line remains vested in the program, except during the relatively small proportion of time that the terminal is in text mode and is actually sending data. If the terminal stops sending data for a period of about 25 seconds, a time-out function in the terminal returns it to control mode. When this happens, the program is again able to initiate activity on the communication line, and the terminal begins monitoring the line for control signals from the computer. This control scheme prevents one terminal on a line from monopolizing use of the line so that the computer is unable to communicate with other terminals on the same line.

The 2741, on the other hand, is intended for conversational use: there is only one terminal per line, and input by the terminal operator and response by the computer alternate, as in an ordinary telephone conversation, until the terminal operator chooses to end the conversation. Each time the terminal is in transmit state, the terminal operator has control; that is, the program can initiate no new activity on the line until the terminal operator returns control to the program by sending an EOT character. The 2741 has no time-out function by which control can be returned to the computer.

Only the terminal operator can end a conversation, either by switching the terminal mode switch to Local or by turning off the terminal power switch.

The sequence of operations between the time the terminal operator begins and ends the conversation with the computer depends upon the logical structure of the program and upon the communications conventions established between terminal operator and the program. These in turn depend on the system application.

Although the differences between line control for the 2741 and for other start-stop terminals result in dissimilar channel programs, you code your READ and WRITE macro instructions in the same way as for other terminals, with the exception of the "entry" operand. Because 2741s use no terminal lists, "entry" is not used, and if coded, it is ignored.

----

*See the IBM 2740/2741 Operator's Guide, for detailed dial-up procedures.

Channel Commands for the IBM 2741

The functions of the commands comprising the channel programs for the 2741 are given below. (The commands are described as they apply to Read and Write operations for the 2741; no inferences should be drawn as to their applicability for other terminals.)

Write EOT	Sets the terminal to transmit state.
Write EOA	Sets the terminal to receive state.
Inhibit	(1) Receives text from terminal into input area. (2) Receives text from terminal but does not place it in input area. Used for purging the communication line of unneeded text data (used in Read Skip channel program).
Enable	For switched lines, conditions the TCU to accept calls from terminals.
Disable	When Disable is the first command of Read Initial or Write Disconnect (switched line), it disables the line if, through program logic error, the line is in the enabled condition when the current channel program is started. If the line is already in the disabled condition, which is the normal case, the Disable command has no effect.
Prepare	For switched lines, causes the TCU to monitor the line for incoming data.

The function of an Inhibit command, like that of a Read command, is to receive data from a terminal. The difference is that a Read command is terminated by expiration of a timeout interval (if not terminated sooner by receipt of data), while an Inhibit is not ended in this way. Channel programs for the IBM 2741 use Inhibit commands, rather than Read commands, because in 2741 operation under BTAM, an indefinite period may elapse between initiation of a channel program and receipt of data from a terminal.

READ MACRO INSTRUCTIONS

READ Initial (TI) (Nonswitched Line)

READ Initial receives message text (beginning with EOA) from the terminal.

1. Prepare (receives EOA)
2. Inhibit (receives text)

READ Initial (TI) (Switched Line)

READ Initial disables the line (in case this was not done previously), enables it, then receives message text (beginning with EOA) from the terminal.

1. Disable
2. Enable
3. Prepare (receives EOA)
4. Inhibit (receives text)

READ Continue (TT) READ Conversational (TV)

READ Continue and READ Conversational are identical operations. Each sets the terminal to transmit state, then receives message text (beginning with EOA) from the terminal.

1. Write EOT
2. Prepare (receives EOA)
3. Inhibit (receives text)

READ Skip (TS)

READ Skip receives message text from the terminal but does not place it in main storage.

1. Inhibit (received text is discarded)

WRITE MACRO INSTRUCTIONS

WRITE Continue (TT)

WRITE Continue sends a message segment to the terminal. It is for use after a WRITE Conversational has set the terminal to receive state.

1. Write Text

WRITE Conversational (TV)

WRITE Conversational sets the terminal to receive state and sends it a message segment. It is for use after a Read operation to reverse the direction of transmission.

1. Write EOA
2. Write Text

WRITE Continue Conversational (TTV)

WRITE Continue Conversational sends the terminal a message segment followed by EOT, which sets the terminal to transmit state, then receives message text (beginning with EOA) from the terminal. WRITE TTV is for use following a WRITE Continue or WRITE Conversational, to reverse the direction of transmission.

1. Write Text
2. Write EOT
3. Prepare (receives EOA)
4. Inhibit (receives text)

WRITE Disconnect (TN)
(Switched Line)

WRITE Disconnect disables the line to break off communication with the terminal.

1. Disable

DESIGNING A MESSAGE CONTROL ROUTINE

This section explains how Read and Write operations may be combined to permit conversational communication between terminal and computer.

The first operation, once the line group has been opened, must be a Read Initial.

When the terminal operator establishes communication, the first line of text he types is read into the input area. The Read Initial ends with receipt of the EOT character sent when the terminal operator presses the Return key or the Attention key.

The program should then determine whether it should receive more text from the terminal. This decision might be based on analysis of the data just received. For example, an operating convention might be established by which the terminal operator presses the Return key to signify that he has further input and the Attention key to indicate that he has finished sending. The Return key causes transmission of the new line (NL) character followed by EOT. The Attention key causes transmission of the EOT character only. The program can check the last two characters received from the terminal to determine the action to take: NL EOT indicating that a Read operation should be executed to receive the next text segment from the terminal operator; EOT alone indicating that the program should reply.

If the program is to reply, it should execute a Write Conversational operation. If desired, you can send the reply in several segments by using Write Continue operations. After the last Write, you should execute a Read Initial (nonswitched) or Read Conversational (switched) operation to permit the terminal operator (1) to resume sending input (in which case the sequence just described is repeated) or (2) to signify to the program that he has finished by turning the mode switch to Local or by turning off the terminal power switch. You may substitute a Write Continue Conversational (TTV) for the last of a sequence of Write Continues, to avoid executing a separate Read Initial or Read Conversational; the WRITE TTV performs the functions of the Write Continue and the subsequent Read.

PROGRAMMING CONSIDERATIONS

Operations on Switched Lines

When the terminal operator switches the terminal to Local mode or turns off the power, the operation in progress at that moment is terminated with an indication of Channel End, Device End, and Unit Check in the CSW status byte, and Intervention Required in the sense byte. These indications will be posted in the DECB for the line as DECSDECB=X'41' and DECFLAGS=X'04'. You should check for the presence of these indicators after each Read and Write operation and take appropriate action; ordinarily, the Read Initial should be reissued.

For some applications it will be appropriate for the terminal operator to end the conversation only when the terminal is in transmit state, that is, not to turn off power while receiving text from the computer. It might then be appropriate for the program to record whether the operation in effect when the terminal went off-line was a Read or a Write.

Operations on Nonswitched Lines

- In a conversational environment, it is usually sufficient that the data is made available to the terminal operator, without the necessity that he actually receives it. If it is imperative that the message be received by the operator, he may be required to acknowledge receipt. The acknowledging message is received via the Read operation that follows the sequence of Write operations.

  If during transmission from computer to terminal, the terminal operator chooses to break off reception of the message, the data set (modem) that connects the terminal to the line cannot signal this fact to the transmission control unit (TCU). When the operator breaks off reception, the sequence of Write operations then in progress proceeds to conclusion just as though the terminal were still receiving. All Write operations are posted complete without error; that is, completion code is X'7F' and DECFLAGS equals zero.

- For half-duplex nonswitched lines, a Read operation is not posted complete until a message has been received. This may mean that your program must accept "sign-on" messages in any Read operation. If it is desirable to recognize the end of a conversation, the program may wait an appropriate amount of time and, if no message has been received yet, assume that the terminal operator has ended the conversation. If necessary to purge the Read operation you may issue an IOHALT macro for the line. This will halt the operation and cause posting of the Read. You may then issue another Read Initial operation, with changes to the area and/or length operands. For example, the new Read operation may be intended to receive a "sign-on" message into a different area than regular messages.

Using the Attention Key and 2741 Interrupt Facility

When the terminal is in transmit state, the operator may press the Attention key to signal the computer that he has finished entering data. Pressing this key sends an EOT to the computer and returns the terminal to receive state. The keyboard locks, and the operator can resume entering data only after the program returns the terminal to transmit state by means of a Read operation or Write Continue Conversational operation.

If the 2741 is equipped with the Interrupt feature, the Attention key can be pressed while the terminal is in receive state to interrupt data transmission from the computer, when, for example, the operator has a high-priority message to enter. Pressing the Attention key causes the Write operation then in progress to terminate, and the Channel End, Device End, and Unit Check indications to be set in the CSW status byte and the Intervention Required indication in the sense byte. These indicators will be posted in the DECB for the line as DECSDECB=X'41' and DECFLAGS=X'02'. You should check after each Write operation for the presence of these indicators. When present, it is generally appropriate to issue a Write Continue that sends a NL character to return the terminal's print element carrier to the beginning of the next printing line.

General Considerations

- Dynamic buffering cannot be used for the 2741.
- The usual considerations regarding use of the WAIT and TWAIT macro instructions should be observed. Before issuing any of these macros, you must always check the return code resulting from a Read or Write operation to ensure that the operation was started successfully.
- Messages sent to a terminal must not contain any EOT characters, as these cause the program to lose control.
- IBM 2741 terminals do not perform an automatic carrier return when the print element reaches the end of the print line. To avoid character pile-up at the end of the line, the text sent to the terminal must contain NL (new line) characters at intervals not exceeding the length (in characters) of the line.
- Any printable characters received by the terminal during the time the terminal is executing a carrier return, horizontal tab, or index (line feed) function will be printed erratically. To avoid this occurrence, each New Line (NL), Horizontal Tab (HT), and Line Feed (LF) character must be followed by one or more nonprinting characters, such as the Idle character.

For the line feed function, you should place one Idle character after each LF character in text to be sent to the terminal. For the new line and tab functions, the number of Idle characters needed equals 1.5 plus the number of inches of carrier travel caused by the function, rounded off to the next higher integer. In addition, you may need to place Idle characters at the beginning of each block of text the program sends to the terminal following receipt of an EOT character from the terminal.

The number of Idles required depends on several factors, such as line turn-around time and model of data set used. A recommended practice is to use the same number of characters as are used following a NL character that results in the longest carrier travel. Example: Assume the length of a print line for a particular application is 7-3/4 inches and tab settings are at 2 and 6 inches. Each HT character should be followed by 1.5 + 4=5.5, or 6 Idles (the 4 derives from the maximum distance of carrier travel [4 inches], caused by an HT character). Each NL character should be followed by 1.5 + 7.75=9.25, or 10 Idles. Also, each block of text sent to a 2741 following receipt of EOT from the terminal should begin with 10 Idles.

AT&T 83B3 SELECTIVE CALLING STATIONS

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an open or wraparound polling list. The list may have one or more terminal entries, each containing a two-character polling sequence (which for the 83B3 is called a Transmitter Start Code).

To define a polling list, code the operand field of a DFTRMLST macro like this:

```
{OPENLST},(xxyy,...)
{WRAPLST}
```

Write Operations

A Write Initial operation requires an addressing list having one or more terminal entries, each containing a two-character addressing sequence (which for the 83B3 is termed a Call Directing Code).

To define an addressing list, code the DFTRMLST macro like this:

```
OPENLST,(xxyy,...)
```

READ MACRO INSTRUCTIONS

READ Initial (TI)

READ Initial successively polls the terminals in the polling list, and upon receiving a positive response to polling, reads a message block.

A single V or M character constitutes a negative response; the message text itself signifies a positive response.

1. Write FIGS H LTRS
2. Write TSC
3. Read Response
4. Read Text

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial addresses a terminal, and if the response to addressing is positive, writes message text to the terminal.

A single V or M character constitutes a positive response; a negative response is indicated by no response at all. A negative response for any terminal in the list is an abnormal condition. The operation ends and is posted complete-with-error.

1. Write FIGS H LTRS
2. Write CDC
3. Write LTRS
4. Read Response
5. Write Text
6. Write FIGS H LTRS (TIR only)

Programming Notes:

1. Each output message must begin with the sequence CR LF LTRS (this serves as the end-of-addressing indicator).
2. You must specify in the WRITE macro the exact length of the message.
3. If you are sending a message with a WRITE TI macro, code FIGS H LTRS at the end of the message (this is the end-of-transmission sequence). If you are sending a message with a WRITE TIR macro, the macro supplies the FIGS H LTRS sequence.

TERMINAL-TO-TERMINAL OPERATION

BTAM does not provide control for terminal-to-terminal traffic on a line on which BTAM provides control of traffic between computer and terminal; however, BTAM does not interfere with terminal-to-terminal traffic. In a system in which such traffic can occur, the operation is as follows.

A READ Initial macro polls the terminal that will become the sending terminal. The sending terminal responds with the addressing code of the terminal with which it wishes to communicate. This code appears to the Read Response command like data, and is therefore received in the input area. The next character is a V or M sent by the receiving terminal as a positive response. It, too, is read into the input area. The sending terminal recognizes the V or M as a positive response and sends a message to the receiving terminal; this message text, too, is read into the input area. Thus, while BTAM does not influence the terminal-to-terminal operation, it does receive into main storage any message sent between the terminals.

WU MODEL 33/35 TWX TERMINALS

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an answering list containing a sequence of control and identification characters to be sent to a terminal that calls the computer. The sequence has from 7 to 18 characters. A recommended sequence is:

Null CR LF DEL (1 to 12 graphic characters) CR LF XON

To code an answering list, code the operand field of a DFTRMLST macro like this:

```
IDLST,0,numsent,sentchar
```

Example: To define an answering list containing the foregoing character sequence (using RALEIGH as the graphic sequence), you would code:

```
IDLST,0,14,01B150FF4B8233A393E212B15088
```

The characters following the third comma are the hexadecimal representations of the transmission code bit patterns for the recommended sequence:

01B150FF --	Null CR LF DEL
4B8233A393E212 --	R A L E I G H
B15088 --	CR LF XON

This sequence prints the computer identification, RALEIGH, at the beginning of the next line, and turns on the tape transmitter.

Read Conversational Operation

A Read Conversational operation requires a list containing a sequence of control characters to be sent to the terminal to prepare it to transmit. For this purpose you define an answering list containing the desired characters; the list is not used for the answering function. Define the list by coding the DFTRMLST operand like this:

```
IDLST,0,numcnsent,cntrlseq
```

If the Read Conversational operation is preceded by a Write Initial operation, a recommended sequence is XON (1 to 4 characters of your choice) XOFF; if the preceding operation was a Read Initial, the single character, XON, may be used. These sequences start the tape transmitter of the terminal. If you wish to read from the keyboard, an appropriate sequence is G Bell A Bell; GA means go ahead, and the bell alerts the terminal operator.

Write Operations

A Write Initial operation requires a calling list containing the same sequence of characters as the called terminal sends when it answers the call from the computer.

To define a calling list, code the operand field of a DFTRMLST macro like this:

```
IDLST,dialcount,dialchars,numsent,tidseq
```

Example: To define a calling list for a terminal whose telephone number is 887-4444 and which will answer with the sequence

CR LF I B M 3 5 A S R # 1 CR LF XON

you would code:

```
IDLST,7,8874444,17,B1509342B205CCAC82CA
4B05C58DB15088
```

READ MACRO INSTRUCTIONS

Programming Note: Each message sent from the terminal (i.e., an incoming message) must end with either the WRU, XON, or XOFF character, or with the EOT sequence. If it ends with the EOT sequence, the next operation must be a Read Initial or Write Initial (EOT resets the terminals to control mode). If the message ends with WRU, XON, or XOFF, the next operation can be a Read Conversational or Write Conversational.

READ Initial (TI)
READ Initial and Reset (TIR)

READ Initial answers a call from a terminal, writes the identification and control sequence, and reads a message block from the terminal.

1. Disable
2. Enable
3. Write Pad characters
4. Write ID-control sequence
5. Read Text
6. Write EOT (TIR only)
7. Disable (TIR only)

READ Conversational (TV)
READ Conversational and Reset (TVR)

READ Conversational writes a control sequence to the terminal and reads message text from the terminal. This macro is for use following a READ Initial or a WRITE Conversational when the line connection is already established.

1. Write Control sequence
2. Read Text
3. Write EOT (TVR only)
4. Disable (TVR only)

WRITE MACRO INSTRUCTIONS

Programming Note: Outgoing messages should not end with a control character or a sequence of control characters (e.g., XON, or XON (user-selected characters) XOFF.

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial calls a terminal and reads the identification sequence of the terminal. If the received ID matches the expected ID that is contained in the terminal list, the macro writes message text to the terminal. If the two ID's do not match, the operation is posted complete-with-error; the message text is not sent.

1. Disable
2. Dial Dial Digits
3. Read Terminal ID sequence
4. Write Text
5. Write EOT (TIR only)
6. Disable (TIR only)

WRITE Conversational (TV)
WRITE Conversational and Reset (TVR)

WRITE Conversational writes message text to the terminal. This macro may be used following a Read operation, to change from receiving text to sending text, and may be issued as many times in succession as necessary to send a message.

1. Write Text
2. Write EOT (TVR only)
3. Disable (TVR only)

WRITE Disconnect (TN)

WRITE Disconnect breaks the line connection.

1. Write EOT
2. Disable

WESTERN UNION PLAN 115A OUTSTATIONS

DEFINING TERMINAL LISTS

Read Operations

A Read Initial operation requires an open or wraparound polling list. The list may have one or more terminal entries, each containing a two-character polling sequence. The first character is always an X (X'17' is the transmission code bit pattern); the second identifies the terminal.

To define a polling list, code the operand field of a DFTRMLST macro like this:

```
{OPENLST},(xxyy,...)
{WRAPLST}
```

Write Operations

A Write Initial operation requires an addressing list having one or more terminal entries, each containing a two-character addressing sequence. The first character is the circuit call code; the second identifies the terminal that is to receive the output message.

To define an addressing list, code the DFTRMLST operand field like this:

```
OPENLST, (xxyy,...)
```

READ MACRO INSTRUCTIONS

READ Initial (TI)

READ Initial successively polls the terminals in the polling list, and upon receiving a positive response to polling, reads a message block.

A single V or M character constitutes a negative response; the message text itself signifies a positive response.

1. Write FIGS H LTRS
2. Write Polling sequence
3. Read Response
4. Read Text

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial addresses a terminal, and if the response to addressing is positive, writes message text to the terminal.

A single V or M character constitutes a positive response; a negative response is indicated by no response at all. A negative response for any terminal is an abnormal condition; the operation ends and is posted complete-with-error.

1. Write FIGS H LTRS
2. Write Addressing sequence
3. Read Response
4. Write Text
5. Write FIGS H LTRS (TIR only)

Programming Notes:

1. Each output message must begin with a Space character (this serves as the end-of-addressing character).

2. You must specify in the WRITE macro the exact length of the message.

3. If you are sending a message with a WRITE TI macro, code FIGS H LTRS at the end of the message (this is the end-of-transmission sequence). If you are sending a message with a WRITE TIR macro, the macro supplies the FIGS H LTRS sequence.

TERMINAL-TO-TERMINAL OPERATION

BTAM does not provide control for terminal-to-terminal traffic on a line on which BTAM provides control of traffic between computer and terminal; however, BTAM does not interfere with terminal-to-terminal traffic. In a system in which such traffic can occur, the operation is as follows.

A READ Initial macro polls the terminal that will become the sending terminal. The sending terminal responds with the addressing code of the terminal with which it wishes to communicate. This code appears to the Read Response command like data, and is therefore received in the input area. The next character is a V or M sent by the receiving terminal as a positive response. It, too, is read into the input area. The sending terminal recognizes the V or M as a positive response and sends a message to the receiving terminal; this message text, too, is read into the input area. Thus, while BTAM does not influence the terminal-to-terminal operation, it does receive into main storage any message sent between the terminals.

WORLD TRADE TELEGRAPH TERMINALS

GENERAL INFORMATION

The name World Trade (WT) telegraph terminals refers to various European teletypewriters using a start-stop 5-level code with two shifts (lettershift and figureshift) to transfer data over point-to-point telegraph lines.

WT terminals use either the International Telegraph Alphabet No. 2 or the Figure Protected Code ZSC3. World Trade telegraph terminals employ the contention system of line control. When a terminal and the computer each try to send a message, simultaneously, both transmissions are immediately stopped; this is called contention.

A terminal is always ready to receive or to send a message. Normally, the motor of the terminal is off and the first lettershift character (LTRS) sent or received by the terminal starts the motor, which requires 1.5 seconds to reach operating speed. During this period, the terminal cannot correctly send or receive characters. The motor stops when no character has been transmitted during a period of from 10 to 30 seconds. When the terminal is operating in this manner, it is said to be in Motor-Off mode. Optionally, the terminal can be equipped with a heavy-duty motor which is never switched off; in this case, the terminal is said to be operating in Motor-On mode.

When a terminal is operating in Motor-Off mode, the MONDLY parameter of the DCB macro instruction enables you to specify the number of Mark (Idle) characters corresponding to the 1.5 second period. When you issue a WRITE macro instruction, BTAM recognizes the motor mode of the terminal (motor-off or motor-on) and generates a LTRS character (this can be followed by a user-specified number of Mark characters) that precedes the data to be sent over the line.

Most terminals can be equipped with another optional feature called the Automatic Answerback Unit. This feature enables a sequence of up to 20 identification characters, generated by a mechanical drum, to be sent over the line by either pressing the IAM key or receiving code combination 4 in figures shift.

Telegraph Adapter Description

The World Trade Telegraph Adapter in the TCU recognizes two message end conditions: FIGS x and FIGS y LTRS. These are established when the IBM 2701, 2702 or 2703 to which the WT terminal is connected is installed: x and y are assigned by the customer on a per-system basis, as follows.

When a terminal is equipped with the Automatic Answerback Unit, FIGS x must be code combination 4 (FIGS D) sent by the terminal WRU key. This character is referred to as the WRU signal. If the terminal is not equipped with the Automatic Answerback Unit, FIGS x may be any other code combination.

The two characters, x and y, cannot be the same. FIGS y immediately followed by a LTRS character causes a Read operation to end. Therefore, FIGS y can be sent by a terminal as data only if it is not followed by LTRS.

The above terminations of a Read operation can be used as end-of-message (EOM) signals. The FIGS y LTRS termination (if not yet used as an EOM signal) or two consecutive EOM signals can represent the end-of-transmission (EOT) signal.

The transmission control unit deletes all incoming LTRS and FIGS characters and updates a shift bit (S) which is added to each character transferred to main storage. Conversely, each change in shift bit setting along a character sequence causes the TCU to send a LTRS or FIGS character ahead of the first message character for which the shift bit was reversed.

Figure 20 shows the relationship of a System/360 byte and a telegraph character configuration.

0	1	2	3	4	5	6	7
(System/360-byte configuration)							
-	-	S	1	2	3	4	5
(Telegraph-character configuration)							

Figure 20. WT Telegraph Code

Contention Resolution

When contention occurs, BTAM sets a completion code of X'7F' in the ECB and turns on bit 3 of DECFLAGS. Contention is resolved by the user program coding and the local operator's action, according to one of the following procedures:

If priority is to be given to the computer, the terminal operator must wait; the program should repeat the Write (or Read TE) operation.

If priority is to be given to the terminal, the program must follow with a READ Continue macro the operation during which contention occurred. The terminal operator continues sending his message.

DEFINING TERMINAL LISTS

In World Trade telegraph operation terminal lists are used only for the READ Continue with Identification Exchange (TE) macro. (They are not used for READ Initial operations.)

If The Terminal's Identification Sequence is to be Requested

To define a terminal list, code the operand field of a DFTRMLST macro like this:

```
WTTALST,0,numrec,ridsent,numsent,tidseq
```

If The Computer Identification is to be sent to the Terminal

```
WTTALST,0,0,0,0,numsent,tidseq
```

Programming Note: tidseq may specify from 7 to 20 characters (computer identification sequence).

READ MACRO INSTRUCTIONS

Read Initial (TI)

READ Initial monitors the line for a message from a terminal, and reads it into the input area. The Read operation ends when an EOM, EOT, or WRU character is received.

1. Prepare
2. Sense
3. Read Text

Programming Note: A RESETPL macro is effective only if issued when message transmission is not in progress.

READ Continue (TT)

READ Continue reads message text from a terminal following receipt of an EOM character, or when the terminal is given the right to transmit when contention has occurred. The operation ends when an EOM, EOT, or WRU is received.

1. Read Text

READ Continue with Identification Exchange (TE)

READ Continue with Identification Exchange writes to the remote terminal the computer's identification sequence (defined in the terminal list) and a WRU character. The operation also reads the identification sequence of the terminal (and optionally, message text) into the input area, only if you code WRU=YES in the DCB macro for the line group.

1.	Write	Mark characters	Note 1
2.	Write	Computer identification seq.	
3.	Write	WRU (or LTRS)	Note 2
4.	Read	Terminal identification	Note 3
5.	Read	Text	

Note 1: One LTRS character plus n Mark characters are sent, where n represents the number of Mark characters, as follows:

a. When the terminal is equipped with the Motor-On optional feature, n is always zero.

b. When the terminal is not equipped with the Motor-On optional feature, n can take one of the following values:

n=0
if the previous operation was a Write, or if a Read operation ended with EOM or WRU signal.

n=the value given to the MONDLY keyword operand of the DCB macro instruction.

Note 2: The computer sends the WRU signal to ask for the terminal identification, provided that WRU=YES is coded in the DCB macro instruction. Otherwise, the computer sends a LTRS character.

Note 3: The terminal sends its own identification. If the received ID and the expected ID do not match, the operation is posted as complete; no message text is read. Bit 3 of DECFLAGS is set to 1.

Programming Note: The value of the length parameter of the READ macro instruction must equal or exceed the length of the identification sequence generated by the Automatic Answerback Unit. If equal, only the terminal identification sequence is sent to the computer. If the length parameter exceeds the identification sequence length, message text can also be read. This is applicable when WRU=YES is specified in the DCB macro instruction; otherwise, command (4) is not generated.

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)

Write Initial sends an output message, preceded by 12 LTRS characters

1.	Write	Mark characters	Note
2.	Write	Pad characters	
3.	Write	Message	

Note: (See Note 1 under READ TE above.)

WRITE Continue (TT)

WRITE Continue sends an output message.

1.	Write	Mark character	Note
2.	Write	Message	

Note: (See Note 1 under READ TE above.)

PROGRAMMING CONSIDERATIONS FOR USE OF AUTO POLL (START-STOP)

Read Initial operations on lines for which the Auto Poll feature is used require polling lists different from those used in programmed polling. They are referred to as Start-Stop Auto Poll lists. The list may be of the open type (SSALST) or wraparound type (SSAWLST), and may have one or more terminal entries, each containing a single polling character (IBM 1030) or a two-character polling sequence (other terminal types). They are specified in the same way as in lists for programmed polling, with the exception of the 2740. The second polling character in a list for the 2740 must be Space.

To define an Auto Poll polling list, code the operand field of a DFTRMLST macro like this:

```
{SSALST }
{SSAWLST},(xx,...)                 (for 1030)
```

```
{SSALST }
{SSAWLST},(xxyy,...) (for 1050,1060,2740)
```

CHANNEL PROGRAMS

Read Initial operations (and Read Continue and Read Repeat operations, on the 1030 and 1060) using Auto Poll require channel programs different from those used in programmed polling. They are explained below by referring to the use of the specific commands that form the program.

READ Initial (TI)
READ Initial and Reset (TIR)
(Open Auto Poll List)

The channel program is:

1. Write EOT EOT EOT
2. Poll
3. NOP
4. Read Index
5. Read Text
6. Write EOA EOT EOT EOT (TIR only)

Command (1) sets the terminals on the line to control mode, as with programmed polling. Command (2) initiates the polling operation. Command (3) is executed only if no response is received from a terminal in the list, or if no terminal in the list returns a positive response to polling, that is, all terminals send negative responses. Execution of command (3) ends the Read operation, which is posted complete in the event control block.

If some terminal in the list returns a positive response, command (3) is skipped; command (4) reads into the first two bytes of the input area the index byte indicating which terminal responded, and the first message character. Command (5) reads the remaining message text into the input area.

Programming Notes:

1. To determine which terminal responded, examine the index byte. You should obtain this index byte not from the input area but from the DECPOLPT field of the DECB for the line. DECPOLPT always contains the index byte, while an I/O error during transmission may prevent the index byte from being placed in the input area.
2. In specifying the length in the READ macro, be sure it is at least one greater than the expected text length, in order to accommodate the index byte.

READ Initial (TI)
READ Initial and Reset (TIR)
(Wraparound Auto Poll List)

The channel program is:

1. Write EOT EOT EOT
2. Poll (Beginning with entry specified in macro)
3. TIC (to command 5)
4. TIC (to command 7)
5. Poll (beginning with first entry in list)
6. TIC (to command 5)
7. Read Index
8. Read Text
9. Write EOA EOT EOT EOT (TIR only)

Command (1) sets the terminals on the line to control mode. Command (2) initiates the polling operation, beginning with the terminal specified by the "entry" operand in the READ macro. If before the end of the list is reached a positive response is returned, the status modifier is set, causing the next command, (3), to be skipped; command (4) transfers to command (7), followed by (8), which functions like commands (4) and (5) in the "open-type" Auto Poll operation.

If, however, the end of the list is reached and no positive response has been received, command (3) is executed, giving control to command (5), which restarts the polling operation at the beginning of the polling list. Polling proceeds automatically, and each time the end of the list is reached, command (6) gives control to (5), and the polling starts again. If during a pass through the list, a positive response is received, command (6) is skipped (just as command (3) is skipped above), and commands (7) and (8) are executed as before.

Programming Notes: The same programming notes given above apply to Auto Poll operations with a wraparound list.

Other Types of READ and WRITE

With two exceptions, all other types of READ macro (such as READ Continue) and all types of WRITE macro generate the same channel programs as are shown under the corresponding type in the appropriate sections for the type of terminal concerned: IBM 1030 Data Collection System, IBM 1050 Data Communication System, IBM 2740 with Station Control Feature, and IBM 2740 with Station Control and Checking Features. The exceptions are READ Continue and READ Repeat for the IBM 1030 and 1060, as shown below. (READ Continue and READ Repeat for the 1050 and for the 2740 with Station Control and Checking are the same as for the non-Auto Poll operations for these terminals; READ Continue and READ Repeat are not provided for 2740 with Station Control and without Checking.)

READ Continue (TT) (1030,1060)

The channel program for READ Continue is identical to the program for READ Initial (using either SSALST or SSAWLST), except that the first command is:

1. Write Y EOT EOT EOT

This channel program sends a positive response, then repolls the terminal and receives message text, as in a Read Initial operation.

READ Repeat (TP) (1030,1060)

The channel program for READ Repeat is identical to the program for READ Initial (using either SSALST or SSAWLST), except that the first command is:

1. Write N EOT EOT EOT

This channel program sends a negative response, then repolls the terminal and receives message text, as in a Read Initial operation.

GENERAL INFORMATION

TRANSMISSION CODES

Binary synchronous communications under BTAM control uses one of three transmission codes, as follows:

System/360 to System/360 (including Model 20), System/3, 1800, 2770, or remote 3270	EBCDIC or USASCII
System/360 to 1130:	EBCDIC
System/360 to 2715:	EBCDIC (transparent)
System/360 to 2780:	EBCDIC, USASCII or Transcode
System/360 to 2972:	EBCDIC

Only EBCDIC may be used between System/360 and a 2770 or 2780 when messages are sent in transparent mode. Only EBCDIC may be used if the central System/360 is communicating with a remote System/360 that is running under BOS (Basic Operating System) or BPS (Basic Programming System). These codes are shown in Appendix E.

You must sometimes enter into message output areas certain line control characters in their USASCII or TRANSCODE form; they must appear in main storage according to the following rules.

- In main storage, bits 1-7 in a System/360 byte correspond to bits b_7-b_1, respectively, of the USASCII character. The zero-bit is always zero (OFF). When the control unit receives a byte, a parity bit is sent over the line along with bits 1-7 of the byte. Conversely, when 7 bits plus a parity bit are received by the transmission control unit from the line, the 7 (data) bits are read into main storage right-justified in a byte and the zero-bit is set to zero.
- For TRANSCODE, a similar rule holds. The hexadecimal equivalent is right-justified in a System/360 byte (bits 2-7) and the 0-bit and 1-bit are always set to zero (off). Only bits 2-7 are sent over the line.

REMOTE STATION COMPATIBILITY AND INTERMIXING

Unlike start-stop terminals, BSC stations of different types are compatible in use of line control procedures, so that it is unnecessary to specify at system generation time what specific type or types of remote station are connected to a given communication line. Instead, one of the three types of line supported by BTAM is coded in the UNIT operand of the system generation IODEVICE macro:

- BSC1 indicates that the line is a non-switched point-to-point line.
- BSC2 indicates that the line is a switched point-to-point line.
- BSC3 indicates that the line is a non-switched multipoint line.

In earlier releases of BTAM, a value representing a specific station type had to be coded in the UNIT operand. For Release 19 of the Operating System, it is still permissible for the UNIT operand to specify station types: S360, 2020 (S/360 Model 20), 1130, and 2780. (See Appendix D for more information). For lines to which an IBM System/3, 2715, 2770, 2792, remote 3270, or 1800 is to be connected, however, the UNIT operand must specify one of the values BSC1, BSC2, or BSC3. For releases of the operating system subsequent to Release 19, only BSC1, BSC2, and BSC3 will be valid UNIT parameters for a binary synchronous line.

USER PROGRAM ANALYSIS

As discussed under Message Transmission in the Line Control and Message Transmission chapter, the user program must analyze the results of each Read or Write operation to determine whether it completed successfully or unsuccessfully, and what if any exceptional condition occurred. The User Program Analysis Procedure section of the Error Recovery Procedures and Error Recording chapter describes a procedure to follow. In addition, the chapter, Suggested Retry Options for BSC Read and Write Operations, recommends appropriate READ and WRITE macro instructions to issue following various error and exceptional conditions.

LINE AND MESSAGE CONTROL FUNCTIONS

ID Verification

Identification sequences may be exchanged between the central computer and some kinds of remote BSC stations with which communication has been established over a switched line. This facility affords either or both stations (i.e., central computer and remote station) the opportunity to verify the identity of the other before message text is transmitted. The terminal list associated with the READ or WRITE macro instruction that established the contact contains the ID sequence to be sent to the remote station, and one or more ID sequences that will be accepted from the remote station.

ID verification is available at either of two levels, which may be termed "regular" and "expanded". In regular ID verification, only one unique ID sequence can be accepted from the remote station, regardless of which of many stations has called (or been called by) the central computer. Further, BTAM makes only one decision regarding continuance of the Read or Write operation. That is, if the received sequence matches the expected sequence (the terminal list contains only one expected sequence), the operation continues, resulting in transfer of text between the stations. If the received sequence does not match the expected sequence, the operation is halted, and text transfer does not occur.

In the expanded ID verification, the user can designate, in the terminal list, many different ID sequences, any of which will be accepted from the remote station; this allows each station to send a unique sequence. Also, contact can be established with stations that do not send ID sequences as well as with those that do. For expanded ID verification, a terminal list having multiple entries is used; this type of list is designated as SWLST. Each entry has a field containing a valid ID sequence that will be accepted from a remote station, and a control byte. (Each entry may also have a user-data area, at the user's option. This is discussed below.)

After the line connection has been established and an ID sequence (or other data) has been received from the remote station, BTAM scans the terminal list for a matching ID sequence. If one is found, BTAM places the address of the entry containing the sequence in the first fullword of the terminal list, for possible use by the user program. Typically, the program would use this address to determine which remote station called or answered the central computer.

The control byte of an entry contains a user-specified indicator specifying what action BTAM is to take after the ID sequence (or other data) has been received. Examples of actions following a Read Connect operation are: continue with the remainder of the Read Connect operation to read a message block; disconnect the line; or post the operation as complete, without reading a message block.

By setting up the control byte prior to the Read or Write operation, and by checking completion codes and indicators in the DECB following receipt of an ID sequence (or other data) from a remote station, the user program can both determine the status of the operation and influence subsequent BTAM actions.

Each terminal list entry may contain a four-byte user-data field. In this field may be placed a relocatable expression as an address that is to be associated with the ID sequence (or ENQ character) contained in that entry. Typically, the user-data field would contain the address of a subroutine to be called when the remote station represented by the ID establishes contact with the central computer.

For more detailed information on use of expanded ID verification, see the descriptions of the READ Connect, WRITE Connect, DFTRMLST, and CHGNTRY macro instructions.

Error Information Byte (EIB) Mode

BTAM provides the option of specifying, in the DCB macro, whether the TCU is to operate in EIB mode or non-EIB mode. The distinction is as follows: In EIB mode, the TCU, during a receive operation, sends an error information byte into main storage following each IUS (US), ETB, and ETX character received from the communication line. In non-EIB mode, the TCU does not send the EIB into main storage following these characters.

The EIB indicates the presence of either a data check or an overrun error (or no error at all) in the sub-block that immediately preceded the IUS (US), ETB, or ETX character. BTAM does not analyze EIBs. The user program may check them and, where an error is found, take appropriate action, such as issuing a READ Repeat with Leading Graphics macro instruction to request retransmission of that part of the message block that is in error.

Whether or not the TCU is operating in EIB mode, it recognizes the IUS (EBCDIC) or US (USASCII) character as signifying the end of an intermediate block. (IUS is Interchange Unit Separator [an EBCDIC character], and US is Unit Separator [a USASCII character]; the two are equivalent characters.)

Double Addressing (Multipoint Lines)

Transient conditions such as lightning impulses or switching pulses can introduce errors in data transmitted over a communication line. Often, such errors consist of inverted bit settings within the bit pattern representing a character. While errors of this kind occurring in message data are normally detected through checking techniques, they are undetected when they occur in polling and addressing (selection) sequences, which are unchecked. An error wherein one valid polling or addressing character is changed to another can result in polling or addressing the wrong station.

To avoid such an occurrence, double addressing may be employed for certain BSC stations. In this technique, a remote station is represented by two identical characters, rather than one character as in single addressing.

When polled or addressed, the remote station that recognizes the first character compares it with the second. If the two are identical, the station address is presumed to be correct, and the station returns a positive response. If they differ, a transmission error is presumed to have altered one or both of the characters, and the station does not return a response.

The increased polling and addressing reliability this technique affords stems from the improbability that both of the characters would be changed in precisely the same way by a transmission error. For example, the characters BB are far less likely to be converted by an error to CC than they are to be converted to BC, or KB, or FC[1]. If a station whose address is K was attached to the line, that station would recognize the first character of the erroneous address KB, but would not respond because the two characters did not match.

----

[1]Each of these conversions could result from a single-bit error in each character, where the transmission code is EBCDIC. For example, the letter B, the bit pattern for which is X'C2' (1100 0010), becomes a C (X'C3', 1100 0011) or a K (X'D2', 1101 0010) through a single-bit error.

Thus, a message intended for station B would not be sent to station K instead.

For System/360 Model 20, System/3, 1800, 2715, 2770, 2972, and remote 3270 stations in a multipoint network, double-addressing must be used.

As is always the case in terminal lists, all list entries must have the same length. Therefore, if addresses of different lengths are to be contained in a list (as when single-addressing is used for some stations, double-addressing for others), the shorter addresses must be padded with leading SYN characters so that they are the same length as the longer addresses.

MESSAGE FORMATS

In nontransparent mode, messages appear on the line in the format:

```
┌────┬──//──────┬───────────────┐
│ STX│ (text)   │ETB (or ETX)   │
└────┴──//──────┴───────────────┘
```

The STX (Start of Text) character is required at the beginning of each message block. (SOH may appear at the beginning of the first message block, however.) ETB denotes the end of a message block and ETX denotes the end of the last block of a message. You must supply in the output area the SOH, STX, ETB and ETX characters. In calculating the length to be specified in a WRITE macro, include the STX and ETX in the number of message characters.

Messages in nontransparent mode may not contain line control characters.

In transparent mode, messages appear on the line in the format:

```
┌───┬───┬─//───┬───┬────┐
│DLE│STX│(text)│DLE│ETX │
└───┴───┴─//───┴───┴────┘
```

Transparent mode allows you to include any bit pattern in the message, regardless of whether the bit pattern represents a line control character.

The DLE STX must appear at the beginning of each message block. DLE ETX denotes the end of the message. You must supply the DLE STX in the beginning of the output area. You do not provide the DLE ETX, as each Write operation of the transparent type automatically sends these characters following your text.

When coding a WRITE macro for sending text in transparent mode, the length must include the DLE STX; the length should not include the ending characters, DLE ETX, as these are sent by a separate command.

When you receive a transparent message from a remote station, it has the format:

DLE STX (text) ETB (or)
DLE STX (text) ETX

The DLE preceding the ETB or ETX is removed by the TCU before the message enters main storage.

If you issue any WRITE macro that specifies both conversational operation and use of dynamic buffering, the BUFL operand of the DCB macro for the line group must specify at least 24 bytes.

Use of Line Control Characters

Successful transmission of data between central computer and remote station demands thorough familiarity with line control (data link control) procedures. See the general discussion of this subject under Use of Line Control Characters in the Line Control and Message Transmission chapter.

Use of SOH and STX Characters

Since either an SOH or an STX character appearing at the beginning block of a message resets, but is not included in, the block check character that follows the block, the following practice is recommended. Include as the first character of a heading, following the SOH character, some specific non-control character that is never used as the first character following STX in a nontransparent text transmission. You may use any character other than a data link control character or the percent sign (%). Consistent observance of this rule will prevent the processing of text data as a heading or of a heading as text data owing to a transmission error that changes STX to SOH or vice versa. When a message block is received without error, presence of the specific character identifies the block as heading, while absence of that character identifies it as text.

Coordinating BSC Central and Remote Programs

In order to achieve message transmission between two computers using BSC communication, you must be careful to coordinate the central and remote programs so they remain in step. This requires that you be aware of the responses that are valid for message text and for each control character that may be sent over the line. These are as follows.

RESPONSES

Responses to Message Text

ACK-0 or ACK-1 (Pos. response)	The remote station received the text correctly.
NAK (Neg. response)	The remote station wishes to have the text retransmitted.
WACK (Wait-before-transmit)	The remote station wishes to delay transmission. (The only valid response to WACK is ENQ (or EOT); the central computer cannot continue sending message text, but must send ENQ until the remote station responds with the positive acknowledgment for the last message block it received. The central computer may, however, respond to WACK with an EOT, to end the transmission.)

Note: When a remote 3270 printer has been started, WACK is a positive response.

leading graphics	The remote station is transmitting user-supplied, non-control characters.
EOT	The remote station is aborting reception of the message because of equipment malfunction or (if the remote station is a computer) program error.
DLE EOT	The remote station is aborting reception of the message and is disconnecting the line because of equipment malfunction or (if the remote station is a computer) program error.
RVI	The remote station wishes temporarily to stop receiving text. The user program may continue sending text, however, or may send an EOT, to end the transmission.

Responses to ENQ

ACK-0	1. The remote station is ready to receive text. 2. Positive response to text.
ACK-1	Positive response to text.
WACK	The remote station wishes to delay transmission.
EOT	The remote station does not wish to receive text.
NAK	The remote station did not acknowledge the last transmission.
Message text	Last receipt was text.

Responses to EOT (Switched Line Only)

EOT	The remote station does not wish to transmit but does not wish to disconnect the line.
ENQ	The remote station wishes to transmit text.
DLE EOT	The remote station is going to disconnect the line.

You should pay close attention to the commands within channel programs. Figure 21 is an example of how central and remote channel programs should be matched. This example is for System/360-to-System/360 communication on a nonswitched point-to-point line. It shows only the sequence of Read and Write operations; it omits checking of return and completion codes and omits WAIT or TWAIT macros.

	Prepare	Write ENQ	
	Read ENQ	Read response	WRITE TI
READ TI	Write response	Write text	
	Read text	Read response	
READ TT	Write response	Write text	
	Read text	Read response	WRITE TTV
		Read text	
WRITE TT	Write text		
	Read response	Write sequence	WRITE TW
		Read ENQ	(Note 1)
WRITE TQ	Write ENQ		
	Read response	Write sequence	WRITE TW
		Read ENQ	(Note 1)
WRITE TQ	Write ENQ		
	Read response	Write sequence	WRITE TW
		Read ENQ	(Note 1)
WRITE TQ	Write ENQ		
	Read response	Write response	READ TT
		Read text	
WRITE TTV	Write text		
(Note 2)	Read response	Write graphics	
	Read text	Write response	READ TTL
		Read text	
WRITE TTR	Write text		
	Read response	Write response	READ TT
	Write EOT	Read text	

Notes:

1. The number of times the WRITE TW is executed depends on how long you wish to delay transmission. The WRITE TW must be followed by WRITE TW, READ TT, READ TP, READ TTL, READ TPL, or READ TRV.

2. On a WRITE TTV the response is read into an input area or user-created buffer.

Figure 21. Example of a BSC Message Control Routine

BSC NONSWITCHED POINT-TO-POINT OPERATION

The macro instructions contained in this section may be issued for any of the types of remote BSC stations that can be connected to a nonswitched point-to-point line, except as noted in individual macro instruction descriptions.

Since BSC operations on nonswitched point-to-point lines use contention-type line control, no terminal lists are used.

The channel programs in this section correspond to an IODEVICE macro UNIT operand of BSC1.

READ MACRO INSTRUCTIONS

READ Initial (TI)

READ Initial monitors the line for an ENQ sent by the remote station, writes a positive response, and reads the message block that follows.

1. Prepare
2. Read ENQ
3. Write Response
4. Read Text

READ Initial Inquiry (TIQ)

READ Initial Inquiry monitors the line for an ENQ sent by the remote station.

1. Prepare
2. Read ENQ

READ Continue (TT)

READ Continue writes a positive response to the remote station and reads a message block.

1. Write ACK-0 or ACK-1
2. Read Text

<u>Note</u>: The text received is either message text or an EOT.

READ Continue with Leading Graphics (TTL)

READ Continue with leading graphics functions the same as a Read Continue, but precedes the positive response with leading graphics.

1. Write Leading Graphics
2. Write ACK-0 or ACK-1
3. Read Text

This macro instruction may be issued for any type of remote BSC station except an IBM 2780 using Transcode; however, the IBM System/3, 1800, 2715, and 2770 ignore the leading graphics characters that precede the response. That is, these characters are neither received into core storage (2715) or terminal buffer, nor passed to any output device attached to the station.

READ Repeat (TP)

READ Repeat writes a negative response to the remote station and reads a message block.

1. Write NAK
2. Read Text

READ Repeat with Leading Graphics (TPL)

READ Repeat with Leading Graphics functions the same as a Read Repeat, but precedes the negative response with leading graphics.

1. Write Leading Graphics
2. Write NAK
3. Read Text

This macro instruction may be issued for any type of remote BSC station except an IBM 2780 using Transcode; however, the IBM System/3, 1800, 2715, and 2770 ignore the leading graphics characters that precede the response. That is, these characters are neither received into core storage (2715) or terminal buffer, nor passed to any output device attached to the station.

READ Inquiry (TQ)

READ Inquiry reads an ENQ from the remote station.

1. Read ENQ

READ Interrupt (TRV)

READ Interrupt writes a Reverse Interrupt (RVI) sequence to indicate to the remote station that the central computer wishes temporarily to stop receiving message text; then issues a Read Text command, which will receive from the remote station either an EOT, signifying end of text transmission, or further text. The RVI sequence is equivalent to, and is recognized by the remote station as, the proper alternating positive acknowledgment (ACK-0 or ACK-1).

READ Interrupt is for use in lieu of a READ Continue.

1. Write RVI sequence
2. Read Text

Programming Notes:

1. Receipt of the RVI sequence does not force the remote station to break off message transmission. It is only an indication that the central computer wishes to stop receiving. The remote station may continue sending message text until such time as it wishes to yield to the central computer by sending EOT. The program in the central computer should therefore be arranged to issue READ Continue macros until the remote station does respond with EOT instead of text.

2. The READ Interrupt macro must not be issued more than once during a transmission, as incorrect alternating acknowledgments may result.

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial writes an ENQ to gain use of the line, and if the response to the ENQ is positive (ACK-0), writes message text and reads the response to text. If the response to ENQ is other than ACK-0, the operation is posted complete, with appropriate indicators set in DECFLAGS.

1. Write ENQ
2. Read Response
3. Write Text
4. Read Response
5. Write EOT (TIR only)

Programming Note: This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.

WRITE Initial Transparent (TIX)
WRITE Initial Transparent and Reset (TIXR)

WRITE Initial Transparent functions the same as the Write Initial, except that after writing message text it writes the ending characters that must follow the transparent data.

1. Write ENQ
2. Read Response
3. Write Text
4. Write DLE ETX
5. Read Response
6. Write EOT (TIXR only)

Programming Note: This macro instruction should not be issued for an IBM 2770 or 2780 on a point-to-point line if the 2770 or 2780 requires component selection characters in the message text.[1]

If both component selection and transparent message text transmission are desired, the component selection characters should be sent in a separate message by a Write Initial operation, followed by a Write TTX (or TTE) or Write TTVX operation to send the transparent text.

WRITE Initial Transparent Block (TIE)

WRITE Initial Transparent Block functions the same as WRITE Initial Transparent (TIX) except that it writes DLE ETB instead of DLE ETX following message text.

1. Write ENQ
2. Read Response
3. Write Text
4. Write DLE ETB
5. Read Response

Programming Note: This macro instruction should not be issued for an IBM 2770 or 2780 on a point-to-point line if the 2770

----

[1]Component selection characters are required unless the Job Select switch (2770) or Mode switch (2780) is set for printing or punching (or some other output device, for 2770), in which case the message is printed or punched, regardless of the component specified by these characters.

or 2780 requires component selection characters in the message text.[1]

If both component selection and transparent message text transmission are desired, the component selection characters should be sent in a separate message by a Write Initial operation, followed by a Write Continue (TT) or Write Continue Transparent (TTX) operation to send the transparent text.

WRITE Initial Conversational (TIV)

WRITE Initial Conversational writes an ENQ to gain use of the line, and if the response to ENQ is ACK-0, writes message text and reads a response, which may be the first two characters of a message block, an alternating acknowledgement, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write ENQ
2. Read ACK-0
3. Write Text
4. Read Response
5. Read Text

Programming Notes

1. This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.
2. This macro instruction may be used for all other types of remote BSC stations; however, the IBM 1800 and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

----

[1]Component selection characters are required unless the Job Select switch (2770) or Mode switch (2780) is set for printing or punching (or some other output device, for 2770), in which case the message is printed or punched, regardless of the component specified by these characters.

WRITE Initial Conversational Transparent (TIVX)

WRITE Initial Conversational Transparent writes an ENQ to gain use of the line, and if the response to ENQ is ACK-0, writes message text and the ending characters, DLE ETX, that must follow the transparent data. The macro then reads a response, which may be either the first two characters of a message block or NAK. If the response is message text, the remaining text is read, if not, the operation is posted complete.

1. Write ENQ
2. Read ACK-0
3. Write Text
4. Write DLE ETX
5. Read Response
6. Read Text

Programming Notes

1. This macro instruction may be used for all types of BSC stations (except as noted in 2, below). However, the IBM 1800, 2715, and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).
2. This macro instruction should not be issued for an IBM 2770 or 2780 on a point-to-point line if the 2770 or 2780 requires component selection characters in the message text.[1]

WRITE Continue (TT)
WRITE Continue and Reset (TTR)

WRITE Continue writes message text and reads a response from the remote station.

1. Write Text
2. Read Response
3. Write EOT (TTR only)

WRITE Continue Transparent (TTX)
WRITE Continue Transparent and Reset (TTXR)

WRITE Contine Transparent writes message text and the ending characters, DLE ETX, that must follow transparent data, and reads a response from the remote station.

1. Write Text
2. Write DLE ETX
3. Read Response
4. Write EOT (TTXR only)

WRITE Continue Transparent Block (TTE)

WRITE Continue Transparent Block writes message text and the ending characters, DLE ETB that must follow transparent data, and reads a response from the remote station.

1. Write Text
2. Write DLE ETB
3. Read Response

WRITE Continue Conversational (TTV)

WRITE Continue Conversational writes message text and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write Text
2. Read Response
3. Read Text

Programming Notes:

1. This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.
2. This macro instruction may be used for all other types of remote BSC stations; however, the IBM 1800 and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

WRITE Continue Conversational Transparent (TTVX)

WRITE Continue Conversational Transparent writes message text and the ending character, DLE ETX, and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write Text
2. Write DLE ETX
3. Read Response
4. Read Text

Programming Note: This macro instruction may be used for all types of remote BSC stations. However, the IBM 1800, 2715, and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

WRITE Inquiry (TQ)

WRITE Inquiry writes an ENQ and reads a response. This macro is for requesting the remote station to transmit its last response (ACK-0, ACK-1, NAK, or a conversational text reply).

1. Write ENQ
2. Read Response
3. Read Text

WRITE Wait-Before-Transmit (TW)

WRITE Wait-before-transmit writes a WACK sequence to a remote station and reads an ENQ. The purpose of this macro is to temporarily stop the remote computer from sending. You may issue it in place of READ Continue or READ Repeat, or in response to a conversational-type Write operation executed by the remote station (if a computer). The only valid responses to WACK are ENQ and EOT. You may issue Write TW repeatedly for as long as necessary to delay your regular response.

1. Write WACK
2. Prepare
3. Read ENQ

WRITE Reset (TR)

WRITE Reset writes an EOT to relinquish use of the line. After sending the EOT, the next operation must be an initial-type Read or Write operation, to again gain use of the line.

1. Write EOT

BSC NONSWITCHED MULTIPOINT OPERATION

The macro instructions contained in this section may be issued for any of the types of remote BSC stations that can be connected to a nonswitched multipoint line, except as noted in individual macro instruction descriptions.

The channel programs shown in this section correspond to an IODEVICE macro UNIT operand of BSC3.

DEFINING TERMINAL LISTS

In order to achieve Read and Write operations over multipoint lines, you must define appropriate terminal lists (i.e., polling or addressing lists) and refer to these lists in your initial-type READ and WRITE macro instructions.

See the explanation of the DFTRMLST macro instruction for general information on defining those lists. Given below are the specific coding requirements for multipoint operations.

Each Read Initial operation requires an open or wraparound polling list, and each initial-type Write operation (Write Initial, Write Initial Transparent, etc.) requires an open addressing list.

Polling List

To define a polling list for any type of BSC station or combination of stations on a multipoint line, code the operand field of a DFTRMLST macro like this:

```
{AUTOLST }
{AUTOWLST},(tidseq,...)
```

tidseq defines an entry in the polling list, and consists of between one and seven polling characters, followed by an ENQ character, all of which must be coded as the hexadecimal equivalents of their transmission code bit patterns.

All polling list entries must be the same length. Therefore, if polling sequences of different lengths are to be contained in a list, the shorter sequences must be padded with leading SYN characters so that they are the same length as the longer sequences.

In defining a polling list of either the open (AUTOLST) or wraparound (AUTOWLST) kind, you must code, following the entries for the stations to be polled, an entry of length equal to the others, and containing EOT characters (in hexadecimal equivalent of the transmission code bit patterns). For example, if the entries for the stations each contain five polling characters plus ENQ, the last entry must be coded as six EOT characters.

Addressing List

To define an addressing list for any type of BSC station or combination of stations on a multipoint line, code the DFTRMLST operand field like this:

```
OPENLST,(tidseq,...)
```

tidseq consists of between one and seven addressing characters, followed by an ENQ character, all of which must be coded as the hexadecimal equivalents of their transmission code bit patterns.

All addressing list entries must be the same length. Therefore, if addresses of different lengths are to be contained in a list, the shorter addresses must be padded with leading SYN characters so that they are the same length as the longer addresses.

Note: See the discussion of double addressing in the General Information section at the beginning of this chapter.

READ MACRO INSTRUCTIONS

READ Initial (TI)
(Using Open Polling List [AUTOLST])

READ Initial initiates an Auto Poll operation to cause the TCU automatically to poll each of the stations in the polling list. If a positive response to polling is received from any station, the macro reads into the input area the index byte indicating which station is sending the message, followed by the message block. The first byte of the input area contains the index.

1. Write EOT
2. Poll (at list entry specified in READ macro)
3. NOP
4. Read Index
5. Read Text

Command (1) sets the stations on the line to control mode. Command (2) initiates the polling operation. Command (3) is executed only if no response is received from a station in the Auto Poll list, or if no station in the list returns a positive response to polling, that is, all stations send negative responses. Execution of command (3) ends the Read operation, which is posted complete in the event control block.

If some station in the list returns a positive response, command (3) is skipped; command (4) reads into the first two bytes of the input area the index byte indicating which station responded, and the first message character. Command (5) reads the remaining message text into the input area.

Programming Notes

1. To determine which station responded, examine the index byte. You should obtain this index byte not from the input area but from the DECPOLPT field of the DECB for the line. DECPOLPT always contains the index byte, while an I/O error during transmission may prevent the index byte from being placed in the input area.

2. In specifying the length in the READ macro, be sure it is at least one greater than the expected text length, in order to accommodate the index byte.

READ Initial (TI) (Using Wraparound Polling List [AUTOWLST])

READ Initial initiates an Auto Poll operation to cause the TCU automatically to poll each of the stations in the polling list. If a positive response is received from any station, the macro reads into the input area the index byte indicating which station is sending the message, followed by the message block. The first byte of the input area contains the index.

1. Write EOT
2. Poll (at list entry specified in READ macro)
3. TIC (to command (5))
4. TIC (to command (7))
5. Poll (at beginning of list)
6. TIC (to command (5))
7. Read Index
8. Read Text

Command (1) sets the stations on the line to control mode. Command (2) initiates the polling operation, beginning with the station specified by the "entry" operand in the READ macro. If a positive response is returned before the end of the list is reached, the status modifier is set, causing the next command, (3), to be skipped; command (4) transfers to command (7), followed by (8), which functions like commands (4) and (5) in the "open-type" Auto Poll operation.

If, however, the end of the list is reached and no positive response has been received, command (3) is executed, giving control to command (5), which restarts the polling operation at the beginning of the polling list. Polling proceeds automatically, and each time the end of the list is reached, command (6) gives control to (5), and the polling starts again. If a positive response is received during a pass through the line, command (6) is skipped (just as command (3) is skipped above), and commands (7) and (8) are executed as before.

Programming Notes

The same programming notes given above apply to Auto Poll operations with a wraparound list.

READ Continue (TT)

READ Continue writes a positive response to the remote station and reads a message block.

1. Write ACK-0 or ACK-1
2. Read Text

Note: The text received is either message text or an EOT.

READ Continue with Leading Graphics (TTL)

READ Continue with leading graphics functions the same as a Read Continue, but precedes the positive response with leading graphics.

1. Write Leading Graphics
2. Write ACK-0 or ACK-1
3. Read Text

This macro instruction may be issued for any type of remote BSC station except an

IBM 2780 using Transcode or a remote IBM 3270; however, the IBM System/3, 1800, 2715, and 2770 ignore the leading graphics characters that precede the response. That is, these characters are neither received into core storage (2715) or terminal buffer, nor passed to any output device attached to the station.

READ Repeat (TP)

READ Repeat writes a negative response to the remote station and reads a message block.

1. Write NAK
2. Read Text

READ Repeat with Leading Graphics (TPL)

READ Repeat with Leading Graphics functions the same as a Read Repeat, but precedes the negative response with leading graphics.

1. Write Leading Graphics
2. Write NAK
3. Read Text

This macro instruction may be issued for any type of remote BSC station except an IBM 2780 using Transcode or a remote IBM 3270; however, the IBM System/3, 1800, 2715, and 2770 ignore the leading graphics characters that precede the response. That is, these characters are neither received into core storage (2715) or terminal buffer, nor passed to any output device attached to the station.

READ Inquiry (TQ)

READ Inquiry reads an ENQ from the remote station.

1. Read ENQ

READ Interrupt (TRV)

READ Interrupt writes a Reverse Interrupt (RVI) sequence to indicate to the remote station that the central computer wishes temporarily to stop receiving message text; then issues a Read Text command, which will receive from the remote station either an EOT, signifying end of text transmission, or further text. The RVI sequence is equivalent to, and is recognized by the remote station as, the proper alternating positive acknowledgment (ACK-0 or ACK-1.)

READ Interrupt is for use in lieu of a READ Continue.

1. Write RVI sequence
2. Read Text

Programming Notes:

1. Receipt of the RVI sequence does not force the remote station to break off message transmission. It is only an indication that the central computer wishes to stop receiving. The remote station may continue sending message text until such time as it wishes to yield to the central computer by sending EOT. The program in the central computer should therefore be arranged to issue READ Continue macros until the remote station does respond with EOT instead of text.

2. The READ Interrupt macro must not be issued more than once during a transmission, as incorrect alternating acknowledgments may result.

3. The remote 3270 always responds to the READ Interrupt macro with an EOT. The problem program must determine whether all data was received by checking for an ETX at the end of the previous message block. If an ETB is present instead, all data was not received. If the rest of the data is wanted, the problem program can reread the message.

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI)
WRITE Initial and Reset (TIR)

WRITE Initial addresses a remote station and if the response to addressing is positive, writes message text, then reads the response.

1. Write EOT
2. Write Addressing sequence
3. Read Response
4. Write Text
5. Read Response
6. Write EOT (TIR only)

Programming Note: This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.

WRITE Initial Transparent (TIX)
WRITE Initial Transparent and Reset (TIXR)

WRITE Initial Transparent addresses a remote station, and if the response to addressing is positive, writes message text and ending characters DLE ETX, then reads the response.

1. Write EOT
2. Write Addressing sequence
3. Read Response
4. Write Text
5. Write DLE ETX
6. Read Response
7. Write EOT (TIXR only)

Programming Note: This macro instruction cannot be used to send message text to an IBM 2972 or a remote IBM 3270, because text transmission to these types of stations must always be in nontransparent mode.

WRITE Initial Transparent Block (TIE)

WRITE Initial Transparent Block addresses a remote station, and if the response to addressing is positive, writes message text and ending characters DLE ETB, then reads the response.

1. Write EOT
2. Write Addressing sequence
3. Read Response
4. Write Text
5. Write DLE ETB
6. Read Response

Programming Note: This macro instruction cannot be used to send message text to an IBM 2972 or a remote IBM 3270, because text transmission to these types of stations must always be in nontransparent mode.

WRITE Initial Conversational (TIV)

WRITE Initial Conversational addresses a remote station and if the response to addressing is positive, writes message text and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write EOT
2. Write Addressing Sequence
3. Read Response
4. Write Text
5. Read Response
6. Read Text

Programming Notes

1. This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.
2. This macro instruction may be used for all other types of remote BSC stations. However, the IBM 1800, 2770, and 2972 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

WRITE Initial Conversational Transparent (TIVX)

WRITE Initial Conversational Transparent addresses a remote station, and if the response to addressing is positive, writes message text and the ending characters, DLE ETX, that must follow the transparent data. The macro then reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write EOT
2. Write Addressing sequence
3. Read Response
4. Write Text
5. Write DLE ETX
6. Read Response
7. Read Text

Programming Notes:

1. This macro instruction cannot be used to send message text to an IBM 2972 or a remote IBM 3270, because text transmission to these types of stations must always be in nontransparent mode.
2. This macro instruction may be used for all other types of remote BSC stations. However, the IBM 1800, 2715, and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

WRITE Continue (TT)
WRITE Continue and Reset (TTR)

WRITE Continue writes message text and reads a response from the remote station.

1. Write Text
2. Read Response
3. Write EOT (TTR only)

Programming Note: This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.

WRITE Continue Transparent (TTX)
WRITE Continue Transparent and Reset (TTXR)

WRITE Continue Transparent writes message text and the ending characters, DLE ETX, that must follow transparent data, and reads a response from the remote station.

1. Write Text
2. Write DLE ETX
3. Read Response
4. Write EOT (TTXR only)

Programming Note: This macro instruction cannot be used to send message text to an IBM 2972 or a remote IBM 3270, because text transmission to these types of stations must always be in nontransparent mode.

WRITE Continue Transparent Block (TTE)

WRITE Continue Transparent Block writes message text and the ending characters, DLE ETB, that must follow transparent data, and reads a response from the remote station.

1. Write Text
2. Write DLE ETB
3. Read Response

Programming Note: This macro instruction cannot be used to send message text to an IBM 2972 or a remote IBM 3270, because text transmission to these types of stations must always be in nontransparent mode.

WRITE Continue Conversational (TTV)

WRITE Continue Conversational writes message text and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write Text
2. Read Response
3. Read Text

Programming Notes

1. This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.

2. This macro instruction may be used for all other types of remote BSC stations. However, the IBM 1800, 2770, and 2972 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

WRITE Continue Conversational Transparent (TTVX)

WRITE Continue Conversational Transparent writes message text and the ending character, DLE ETX, and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write Text
2. Write DLE ETX
3. Read Response
4. Read Text

Programming Notes

1. This macro instruction cannot be used to send message text to an IBM 2972 or a remote IBM 3270, because text transmission to these types of stations must always be in nontransparent mode.

2. This macro instruction may be used for all other types of remote BSC stations. However, the IBM 1800, 2715, and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

WRITE Inquiry (TQ)

WRITE Inquiry writes an ENQ and reads a response. This macro is for requesting the remote station to transmit its last response (ACK-0, ACK1, NAK or a Conversational Text reply).

1. Write ENQ
2. Read Response
3. Read Text

WRITE Wait-Before-Transmit (TW)

WRITE Wait-before-transmit writes a WACK sequence to a remote station and reads an ENQ. The purpose of this macro is to temporarily stop the remote computer from sending. You may issue it in place of READ Continue or Read Repeat, or in response to a conversational-type Write operation executed by the remote computer. The only valid responses to WACK are ENQ and EOT. You may issue Write TW repeatedly for as long as necessary to delay your regular response.

1. Write WACK
2. Read ENQ

WRITE Reset (TR)

WRITE Reset writes an EOT to relinquish use of the line. After sending the EOT, the next operation must be an initial-type Read or Write operation, to again gain use of the line.

1. Write EOT

BSC SWITCHED POINT-TO-POINT OPERATION

The macro instructions contained in this section may be issued for any of the types of remote BSC stations that can communicate with the central computer over a switched line, except as noted in individual macro instruction descriptions.

Contact between central computer and a remote station over a switched line can be established in numerous ways, representing the various combinations of these alternatives:

- Is the central computer to call a remote station or answer a call from a remote station?
- Is the calling or answering function to be automatic or manual? That is, are the TCU and the common carrier equipment at the central computer equipped to perform the calling (dialing) or answering function under program control, or must the operator at the central computer perform these functions?
- Is the data set (modem) at the central computer capable of generating and transmitting a "data tone" to signify to a calling station that data transfer can proceed, or must the user program supply the data tone?
- Once contact has been established, is the central computer to send an ID sequence to the remote station, is the central computer to receive an ID sequence from the remote station, or both (or neither)? If ID sequences are to be received from remote stations, do all stations with which contact may be established have to send the same ID sequence? Or can each send a unique sequence?
- Once contact has been established, is the direction of the first message transmission to be toward the remote station (i.e., a Write Text operation), or toward the central computer (i.e., a Read Text operation)?

Each of the various available combinations of the foregoing alternatives is represented by a combination of a specific READ or WRITE macro instruction option and a terminal list having a specific format and content. These are shown in Table 8.

Table 8. Summary of BSC Switched Line READ and WRITE Macro and Terminal List Options

TO:	and to...	and if EXPANDED ID Verification...	to be used issue a...	macro that refers to a terminal list defined like this:
CALL a remote station, using... Automatic Calling	Read text	is not³	READ TI	BSCLST,dialcount, dialchars,numrec, ridseq,numsent,tidseq (See Note 1)
	Write text	is	WRITE TC (followed by WRITE TT)	SWLST,AD,dialcount, dialchars,entrylength, [userlength],idcount idsent[{,(auth-sequence[,control-value][,user-data])}...]
		is not	WRITE TI⁶	BSCLST,dialcount, dialchars,numrec, ridseq,numsent,tidseq
Manual Calling	Write text⁴	is	WRITE TC (followed by WRITE TT)	SWLST,MD,entry-length,[userlength], idcount,idsent [{,(authsequence [,controlvalue] [,userdata])}...]
		is not	WRITE TC (followed by WRITE TT)	WTLIST,0,numrec, ridseq,numsent, tidseq
ANSWER a remote station,using... Automatic Answering	Read text	is	READ TC or TCW	SWLST,AN,entrylength, [userlength], idcount,idsent [{,(authsequence [,controlvalue] [,userdata])}...]
		is not	READ TI	BSCLST,0,numrec, ridseq,numsent, tidseq (see Note 2)
Manual Answering	Read text⁵	is not³	READ TI	(if data set [modem] automatically generates tone) BSCLST,0,numrec, ridseq,numsent,tidseq (if data set [modem] does not automatically generate tone) WTLIST, 0,numrec,ridseq, numsent,tidseq, length,area

¹Alternatively, if no ID sequences are required, the list may be coded as: DIALST,dialcount, dialchars
²Alternatively, if no ID sequences are required, the list may be coded as: DIALST,0
³Expanded ID verification not available.
⁴Text cannot be read from the remote station.
⁵Text cannot be written to the remote station.
⁶Or WRITE TIX, TIV, or TIVX.

DEFINING TERMINAL LISTS

See the explanation of the DFTRMLST macro instruction for general information on defining these lists. Given below are the specific coding requirements for switched point-to-point operations.

Automatic Calling and Answering - With Expanded ID Verification

To define a calling list, code the operand field of a DFTRMLST macro instruction like this:

```
SWLST,AD,dialcount,dialchars,entrylength,
[userlength],idcount,idsent
[{,(authsequence[,controlvalue]
[,userdata])}...]
```

This type of list is for use with a WRITE Connect (TC) macro instruction.

To define an answering list, code the operand field of a DFTRMLST macro like this:

```
SWLST,AN,entrylength,
[userlength],idcount,idsent
[{,(authsequence[,controlvalue]
[,userdata])}...]
```

This type of terminal list is for use with a READ Connect (TC) or Read Connect with Tone (TCW) macro instruction.

See the section DFTRMLST Macro -- SWLST Form, for detailed information on coding DFTRMLST macros of the SWLST form.

Automatic Calling and Answering - Without Expanded ID Verification

To define a calling list, code the operand field of a DFTRMLST macro like this:

```
BSCLST,dialcount,dialchars,numrec,ridseq,
numsent,tidseq
```

The ridseq operand must end with ACK-0; the tidseq operand must end with ENQ. This type of list is for use with a READ Initial or WRITE Initial macro instruction.

To define an answering list, code the DFTRMLST operand field like this:

```
BSCLST,0,numrec,ridseq,numsent,tidseq
```

The ridseq operand must end with ENQ; the tidseq operand must end with ACK-0. This type of list is for use with a READ Initial macro instruction.

Omitting ID Sequence: If no ID sequences are desired, omit, in the ridseq and tidseq operands, all but the ENQ and ACK-0 characters. When no ID characters are specified, the numrec, ridseq, numsent, and tidseq operands are as follows:

	Calling List	Answering List
numrec	2	1
ridseq[1]	ACK-0	ENQ
numsent	1	2
tidseq[1]	ENQ	ACK-0

(Alternatively, if ID sequences are not desired, you may define a calling list using a DFTRMLST in which the operand field is coded DIALST,dialcount,dialchars; an answering list using a DFTRMLST in which the operand field is coded DIALST,0.)

Manual Calling[2] - with Expanded ID Verification

To define a calling list, code the operand field of a DFTRMLST macro like this:

```
SWLST,MD,entrylength,
[userlength],idcount,idsent
[{,(authsequence[,controlvalue]
[,userdata])}...]
```

This type of list is for use with a WRITE Connect (TC) macro instruction.

---

[1]The ridseq and tidseq operands must be coded in hexadecimal representation of the appropriate transmission code bit patterns of the ENQ and ACK-0 characters.
[2]Manual answering with expanded ID verification is not available.

Manual Calling and Answering - Without Expanded ID Verification

To define a calling list, code the operand field of a DFTRMLST macro like this:

```
WTLIST,0,numrec,ridseq,numsent,tidseq
```

The ridseq operand must end with ACK-0; the tidseq operand must end with ENQ. The sequence specified by tidseq may contain up to 15 characters (excluding the ENQ). This type of list is for use with a WRITE Connect macro instruction.

An answering list may be coded in one of two ways, depending on whether the data set (modem) at the answering station (i.e., central computer) is or is not designed to automatically generate a data tone upon receiving a call.

Data Sets Without Tone: If the data set does not generate a tone, BTAM sends a user-specified character sequence that the operator at the calling station hears as an audible tone.

To define an answering list for a line equipped with a data set that does not generate a tone, code the operand field of a DFTRMLST macro like this:

```
WTLIST,0,numrec,numsent,tidseq,length,
area
```

The ridseq operand must end with an ENQ; the tidseq operand must end with ACK-0. The sequence specified by tidseq may contain up to 15 characters (excluding the two-character sequence, ACK-0).

The length and area operands specify the length of the character sequence used as a data tone and the address of that sequence.

The data tone should be about three seconds long. To obtain a tone of this duration requires a length of about 255 characters, for a 600 bits-per-second communication line, or about 450 characters, for a 1200 bps line. A sequence of X'FF' is recommended for the data tone. This type of list is for use with a READ Initial macro.

Data Sets With Tone: To define an answering list for a list for a line equipped with a data set that generates a tone, code the DFTRMLST operand field like this:

```
BSCLST,0,numreq,numsent,tidseq
```

The ridseq and tidseq operands are as explained above (under Data Sets Without Tone). This type of list is for use with a READ Initial macro.

If no ID sequences are desired, omit in the ridseq and tidseq operands all but the ENQ and ACK-0 characters. When no ID characters are specified, the numrec, ridseq, numsent, and tidseq operands are as shown above under Automatic Calling and Answering (Without Expanded ID Verification).

DEFINING TERMINAL LIST (SWLST) FOR EXPANDED ID VERIFICATION

Answering List

A READ Connect macro for Automatic Answering, with Expanded ID Verification, requires an answering list defined as follows:

Name	Operation	Operands
symbol	DFTRMLST	SWLST,AN,entrylength, [userlength],idcount, idsent[{,(authsequence [,controlvalue] [,userdata])}...]

SWLST
: Specifies a list structure for expanded BSC ID verification.

AN
: Specifies that an answering list (to be used by the READ Connect or Read Connect with Tone macro) is to be defined.

entrylength
: Specifies the number of bytes to be allocated for each list entry containing a user-defined authorized ID ENQ sequence. The integer specified should equal the number of bytes required to accommodate the authorized ID ENQ sequence of maximum length, plus the userdata field, if present (4 or 0), plus one (for the entry's control byte). Authorized ID ENQ sequences of less than the maximum length are assembled left-justified within the fixed-length ID field allocated for each entry. Each userdata field (if any) and control byte have the same offset within all entries.

(The value specified may be zero if no other sequence than ENQ, alone, is expected and ENQ is not put in the list.)

userlength
: Specifies whether a four-byte userdata field is to be allocated for each list entry containing an authorized ID ENQ sequence. A code of 4 means to allocate; 0 means not to allocate. The default option is 0.

idcount
: Specifies the length (in bytes) of the field required to accommodate the ID characters (if any) and ACK-0 defined by the idsent operand. The range permitted is 2 (ACK-0 alone) through 17; up to 15 ID characters may be specified.

idsent
: Specifies the hexadecimal representation of the ID ACK-0 sequence to be sent to the remote station. While the ID characters (if any) are of your choosing, the ACK-0 sequence is required. Upon receiving an ID ENQ sequence during execution of a READ Connect, BTAM checks the control byte value of the corresponding list entry, and transmits the ID ACK-0 sequence if the checked value is 0. (See the discussion of the controlvalue suboperand, or the discussion of the Read Connect channel program, for the explanation of the BTAM actions performed for the various control byte values.)

authsequence
: Specifies the hexadecimal representation of an authorized ID ENQ sequence. Each ID ENQ sequence is defined in a separate sublist along with its corresponding control byte value and user data (if any). You should code a separate ID ENQ sequence for each authorized sequence that can be received on a Read Connect operation using the particular answering list being defined. ID ENQ sequences of varying lengths can be defined within the same DFTRMLST macro. Each sequence specified must include the ENQ character at the end. You may define ENQ alone as an authsequence operand to service remote stations not employing ID verification.

controlvalue
: Each list entry assembled for an authsequence sequence has an associated control byte, the value of which determines the automatic BTAM action to be performed when the sequence is received on a Read Connect operation using the list. The values and corresponding BTAM actions are:

0
: Specifies that BTAM is to send the idsent sequence and read a message block (if any) from the calling station. If you omit the controlvalue suboperand within a sublist, this value is assumed. (If the controlvalue operand is omitted, two commas must precede a coded userdata operand in the same sublist, because they are positional operands within the sublist.) You may specify this value for a list entry containing an ID ENQ sequence or the single ENQ character.

1
: Specifies that BTAM is to break the line connection and restart the channel program at the Enable command (to await a new call). You may specify this action if BTAM is not to service a particular calling station at the time of the call. Typically, this action would be specified for reasons of priority (time-of-day scheduling). You may specify this value for a list entry containing an ID ENQ sequence or the single ENQ character.

2
: Specifies that BTAM is to post normal completion of the Read Connect immediately, with the address of the received ENQ character in the first word of the answering list. This permits control to be returned to the user program so that it can specify the subsequent actions to be performed. You may specify this value only for a list entry containing a single ENQ character (i.e., not containing an ID sequence). This permits the user program to issue a subsequent READ Continue or READ Repeat macro to send ACK-0 or NAK to a calling station that is not prepared to receive an ID sequence.

userdata
: Specifies the relocatable expression to be assembled right-justified in the userdata field of the associated list entry. If you omit this suboperand and userlength specifies 4, four noninitialized bytes are allocated for the corresponding list entry. (No boundary alignment can be assumed for the user data field.)

Notes:

1. A maximum of 194 sublists can be coded for an answering list of the SWLST form.

2. The CHGNTRY macro can be used to change the control byte of an answering list of the SWLST form during program execution.

The first fullword of the list is the area in which BTAM stores the address of the entry containing the ID ENQ sequence corresponding to the received sequence. See Appendix A for the format of the assembled answering list.

Calling List

For Automatic or Manual Calling, with Expanded ID Verification, A WRITE Connect macro requires a calling list defined as follows:

Name	Operation	Operands
symbol	DFTRMLST	SWLST,{AD}, {MD} [dialcount,dialchars,] entrylength,[user- length],idcount,idsent [{,(authsequence [,controlvalue] [,userdata])}...]

SWLST
Specifies a list structure for expanded BSC ID verification.

AD
Specifies that an auto-dial calling list is to be defined. In this case, the dialcount and dialchars operands are required so that program-initiated dialing can occur. The corresponding Write Connect channel program begins with a Dial command.

MD
Specifies that a manual-dial calling list is to be defined. In this case, omit the dialcount and dialchars operands, because the dialing operation is initiated by the central computer operator. The Write Connect channel program with which a manual-dial calling list is used begins with an Enable command.

dialcount
Specifies the number of dial characters (bytes) used in the dialing operation. Code this operand only if you code AD as the preceding operand.

dialchars
Specifies the decimal digits of the telephone number to be dialed. Code this operand only if you also code AD.

entrylength
Specifies the number of bytes to be allocated for each list entry containing a user-defined authorized ID ACK-0 sequence. The integer specified should equal the number of bytes required to accommodate the authorized ID ACK-0 sequence of maximum length, plus the userdata field, if present (4 or 0), plus one (for the entry's control byte). Authorized ID ACK-0 sequences of less than the maximum length are assembled left-justified within the fixed-length ID field allocated for each entry. Each userdata field (if any) and control byte have the same offset within all entries.

userlength
Specifies whether a four-byte userdata field is to be allocated for each list entry containing an authorized ID ACK-0 sequence. A code of 4 means to allocate; 0 means not to allocate. The default option is 0.

idcount
Specifies the length (in bytes) of the field required to accommodate the ID characters (if any) and ENQ defined by the idsent operand. The range permitted is 1 (ENQ alone) through 16; up to 15 ID characters may be specified.

idsent
Specifies the hexadecimal representation of the ID characters (if any) and ENQ to be sent to the remote station. Typically, the ID characters to be sent will convey station identification. The ID characters, if any, are of your choosing; the ENQ character is required.

authsequence
Specifies the hexadecimal representation of an authorized ID ACK-0 sequence. Each ID ACK-0 sequence is defined in a separate sublist along with its corresponding control byte value and user data (if any). You should code a separate ID ACK-0 sequence for each authorized sequence that can be received from remote (answering) stations. ACK-0 must be coded following each ID sequence; it must not be coded where no ID sequence is used. BTAM checks for reception of ACK-0 or NAK, alone, on a Write Con-

nect operation without requiring that they appear in the list.

controlvalue
Each list entry assembled for an <u>auth-sequence</u> sequence has an associated control byte. For any received ID sequence terminated by ACK-0, BTAM ignores the control byte. When a valid ID sequence terminated by NAK is received during a Write Connect operation, BTAM examines the control byte of the entry whose ID matches the received ID. The control byte value determines the BTAM action to be performed. The values and BTAM actions are:

0
Specifies that upon receipt of the sequence, BTAM is to post completion of the operation immediately.

1
Specifies that upon receipt of the sequence, BTAM is to resend the ID ENQ sequence. This option has meaning only when the ID NAK sequence has been sent, indicating that the remote station is not ready to receive, and you wish to retry, expecting that the remote station will shortly become ready to receive. The maximum number of retries performed for this control byte value is seven. If more retries than this are desired, you can reissue the WRITE Connect macro; BTAM bypasses the initial Enable or Dial command if the line connection is already established.

userdata
Specifies the relocatable expression to be assembled right-justified in the <u>userdata</u> field of the associated list entry. If you omit this suboperand and <u>userlength</u> specifies 4, four non-initialized bytes are allocated for the corresponding list entry. (No boundary alignment can be assumed for the user data field.)

<u>Notes</u>:

1. A maximum of 192 sublists can be coded for a calling list of the SWLST form.

2. The control byte values for a calling list cannot be changed by use of the CHGNTRY macro.

The first fullword of the list is the area in which BTAM stores (prior to completion posting) the address of the list entry associated with the received ID sequence. See Appendix A for the format of the assembled calling list.

READ MACRO INSTRUCTIONS

<u>READ Initial (TI)</u>
<u>(Using Automatic Calling List -- BSCLST DIALST)</u>

READ Initial calls a remote station, writes the central computer's identification sequence and ENQ to the station, and reads the identification sequence of the remote station and a response. If the identification matches the identification contained in the terminal list, and the response is positive (ACK-0), the operation continues by writing EOT (indicating that the central computer does not wish to send), reading ENQ and responding with ACK-0, then reading a message block from the remote station. If the identifications do not match, the Read operation ends with command (3) and is posted as complete.

1. Dial Dial Digits
2. Write ID ENQ
3. Read ID ACK-0
4. Write EOT
5. Read ENQ
6. Write ACK-0
7. Read Text

<u>READ Initial (TI)</u>
<u>(Using Automatic Answering List -- BSCLST, DIALST)</u>

READ Initial answers a call from a remote station, reads the identification sequence of the remote station and an ENQ, writes ACK-0 to indicate that the central computer is ready to receive, and reads a message block from the remote station.

1. Enable
2. Read ID ENQ
3. Write ID ACK-0
4. Read Text

<u>READ Initial (TI)</u>
<u>(Using Manual Answering List -- BSCLST,WTLIST)</u>

A READ Initial macro using a manual answering list is for use where the central computer is not capable of automatically answering calls from remote stations; the operator at the central computer must answer them manually. Operation is as follows.

The channel program first enables the line so that calls can be received. When the telephone rings, the computer operator answers it, and may verify the identity of the calling station (if that call was initiated by the remote station operator rather than automatically). The operator then places the data set (modem) in data mode. (This terminates the Enable command.)

If the manual answering list is of the WTLIST format (used where the data set (modem) does not automatically generate a data tone), the channel program then sends a user-specified character sequence that the operator at the remote station hears as a tone. If the list is of the BSCLST format (used where the data set does generate a tone), the channel program does not send the character sequence.

The channel program then reads an identification sequence, ending in ENQ, from the remote station. If the sequence does not match the expected sequence, the Read Initial operation ends at this point, and is posted complete-with-error in the event control block. If the two sequences do match, the channel program sends the identification sequence of the central computer, then reads a message block from the remote station.

1.	Enable	
2.	Write	Data Tone Characters (for WTLIST only)
3.	Read	ID ENQ
4.	Write	ID ACK-0
5.	Read	Text

Programming Note: It may be desirable, after issuing the READ Initial, to send a message to the console operator (using the WTO macro), instructing him to answer calls received by the computer.

READ Connect (TC) (Expanded ID Verification) (Using Automatic Answering List - SWLST)

READ Connect is used to allow initial contact to be established with a remote BSC station and to perform a specific action based on the ID sequence, if any, received from the remote station. The possible actions include reading message blocks, disconnecting the line, and immediately returning control to the user program.

After the sequence is received, BTAM analyzes it. If the sequence matches one of the authorized sequences in the answering list, BTAM places the address of the entry containing the matching ID-ENQ sequence (or ENQ alone) in the first fullword of the list. BTAM then examines the control byte of that list entry to determine which action to take.

If the control byte value is 0, BTAM restarts the channel program to send the ID ACK-0 sequence (or ACK-0 alone) given in the list, and then reads a message block, if any. If the control byte value is 1, BTAM restarts the channel program to break the line connection, and then restarts the channel program from the beginning Enable command. If the control byte value is 2, BTAM immediately posts normal completion (X'7F'). (A control byte value of 2 is for use when no ID sequence is employed, and you wish to follow normal completion (X'7F') of the Read Connect operation with a READ Continue macro.)

If the received sequence does not match any of the authorized ID-ENQ sequences (or ENQ alone), BTAM determines whether ENQ alone, an invalid sequence, or DLE EOT was received.

If ENQ alone was received, BTAM posts normal completion (X'7F').

If an invalid sequence was received, BTAM retries the Read ID ENQ command up to seven times. If all retries are unsuccessful, BTAM disconnects the line, turns on bit 3 of DECFLAGS, and posts a completion code of X'7F' (normal completion). If DLE EOT was received, BTAM turns on bit 1 of DECFLAGS and posts normal completion (X'7F').

If a timeout occurs on the Read ID ENQ command, BTAM disconnects the line and restarts the channel program at the Enable command.

This macro is used only when the expanded ID verification facility is to be employed. The entry operand of the READ Connect macro must specify the name of an answering list of the SWLST format, as defined by a DFTRMLST macro. The channel program generated for the READ Connect macro is:

1.	Enable	
2.	Read	ID ENQ (or ENQ alone)
3.	Write	ID ACK-0 (or ACK-0 alone)
4.	Read	Text
5.	Write	DLE EOT
6.	Disable	
7.	TIC to Enable command	

READ Connect with Tone (TCW) (Expanded ID Verification) (Using Automatic Answering List -- SWLST)

READ Connect with Tone functions the same as READ Connect (TC), as described above, except that the channel program contains an added command, Write Data Tone Characters.

This macro is for use on a line equipped with an automatic answering unit that does not automatically send a data tone upon receiving a call. Upon completion of the Enable command, which occurs when a call is received, the channel program sends a user-specified character sequence that the operator at the calling station hears as an audible tone.

The character sequence that constitutes the tone must be coded in the user program. The address and the length of the tone character sequence must be specified in the outarea and outlength operands of the READ TCW macro instruction.

The data tone should be about three seconds long. To obtain a tone of this duration requires a length of about 255 characters, for a 600 bits-per-second communications line, or about 450 characters for a 1200 bps line. A sequence of X'FF' is recommended for the data tone. (Notice that the address and length of the tone sequence are specified in the READ macro, not in the DFTRMLST macro, as is the case for manual answering, without expanded ID verification.)

The channel program generated for the READ Connect with Tone macro is:

1. Enable
2. Write Data Tone Characters
3. Read ID ENQ (or ENQ alone)
4. Write ID ACK-0 (or ACK-0 alone)
5. Read Text
6. Write DLE EOT
7. Disable
8. TIC to Enable command

READ Continue (TT)

READ Continue writes a positive response to the remote station and reads a message block.

1. Write ACK-0 or ACK-1
2. Read Text

Note: The text received is either message text or an EOT.

READ Continue with Leading Graphics (TTL)

READ Continue with leading graphics functions the same as a Read Continue, but precedes the positive response with leading graphics.

1. Write Leading Graphics
2. Write ACK-0 or ACK-1
3. Read Text

This macro instruction may be issued for any type of remote BSC station except an IBM 2780 using Transcode; however, the IBM System/3, 1800, 2715, and 2770 ignore the leading graphics characters that precede the response. That is, these characters are neither received into core storage (2715) or terminal buffer, nor passed to any output device attached to the station.

READ Repeat (TP)

READ Repeat writes a negative response to the remote station and reads a message block.

1. Write NAK
2. Read Text

READ Repeat with Leading Graphics (TPL)

READ Repeat with Leading Graphics functions the same as a Read Repeat, but precedes the negative response with leading graphics.

1. Write Leading Graphics
2. Write NAK
3. Read Text

This macro instruction may be issued for any type of remote BSC station except an IBM 2780 using Transcode; however, the IBM System/3, 1800, 2715, and 2770 ignore the leading graphics characters that precede the response. That is, these characters are neither received into core storage (2715) or terminal buffer, nor passed to any output device attached to the station.

READ Inquiry (TQ)

READ Inquiry reads an ENQ from the remote station.

1. Read ENQ

READ Interrupt (TRV)

READ Interrupt writes a Reverse Interrupt (RVI) sequence to indicate to the remote station that the central computer wishes temporarily to stop receiving message text; then issues a Read Text command, which will receive from the remote station either an EOT, signifying end of text transmission, or further text. The RVI sequence is equivalent to, and is recognized by the remote station as, the proper alternating positive acknowledgment (ACK-0 or ACK-1).

READ Interrupt is for use in lieu of a READ Continue.

1. Write RVI sequence
2. Read Text

Programming Notes:

1. Receipt of the RVI sequence does not force the remote station to break off message transmission. It is only an indication that the central computer wishes to stop receiving. The remote station may continue sending message text until such time as it wishes to yield to the central computer by sending EOT. The program in the central computer should therefore be arranged to issue READ Continue macros until the remote station does respond with EOT instead of text.

2. The READ Interrupt macro must not be issued more than once during a transmission, as incorrect alternating acknowledgments may result.

WRITE MACRO INSTRUCTIONS

WRITE Initial (TI) (Using Automatic Calling List -- BSCLST)

WRITE Initial calls a remote station, writes the central computer's identification sequence and ENQ, and reads the identification sequence of the remote station and a response. If the identification matches the identification contained in the terminal list, and the response is positive (ACK-0), the operation writes message text to the remote station and reads a response. If the identifications do not match, the Write operation ends with command (3) and is posted as complete.

1. Dial Dial digits
2. Write ID ENQ
3. Read ID ACK-0
4. Write Text
5. Read Response

Programming Note: This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.

WRITE Initial Transparent (TIX) (Using Automatic Calling List -- BSCLST)

WRITE Initial Transparent calls a remote station, writes the central computer's identification sequence and ENQ, and reads the identification sequence of the remote station and a response. If the identification matches the identification contained in the terminal list, and the response is positive (ACK-0), the operation writes message text and the ending characters DLE ETX to the remote station, and reads a response. If the identifications do not match, the Write operation ends with command (3) and is posted as complete.

1. Dial Dial digits
2. Write ID ENQ
3. Read ID ACK-0
4. Write Text
5. Write DLE ETX
6. Read Response

Programming Note: This macro instruction should not be issued for an IBM 2770 or 2780 on a point-to-point line if the 2770 or 2780 requires component selection characters in the message text.[1] The reason is that when operating in transparent mode, the 2770 and 2780 do not recognize component selection characters within message text.

If both component selection and transparent message text transmission are desired, the component selection characters should be sent in a separate message by a Write Initial operation, followed by a Write TTVX or Write TTX (or TTE) operation to send the transparent text.

WRITE Initial Transparent Block (TIE)

WRITE Initial Transparent Block calls a remote station, writes the central computer's identification sequence and ENQ, and reads the identification sequence of the remote station and a response. If the identification matches the identification contained in the terminal list, and the response is positive (ACK-0), the operation writes message text, and the ending characters DLE ETB to the remote station, and reads a response. If the identifications do not match, the Write operation ends with command (3) and is posted as complete.

1. Dial Dial digits
2. Write ID ENQ
3. Read ID ACK-0
4. Write Text

----

[1]Component selection characters are required unless the Job Select switch (2770) or Mode switch (2780) is set for printing or punching (or some other output device, for 2770), in which case the message is printed or punched, regardless of the component specified by these characters.

5. Write DLE FTB
6. Read Response

Programming Note: See programming note under Write TIX macro instruction.

WRITE Initial Conversational (TIV) (Using Automatic Calling List -- BSCLST)

WRITE Initial Conversation calls a remote station, writes the central computer's identification sequence and ENQ, and reads the identification sequence of the remote station and a response. If the identification matches the identification contained in the terminal list, and the response is positive (ACK-0), the operation writes message text to the remote station and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

If the identifications do not match, the Write operation ends with command (3) and is posted as complete.

1. Dial Dial digits
2. Write ID ENQ
3. Read ID ACK-0
4. Write Text
5. Read Response
6. Read Text

Programming Notes:

1. This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.

2. This macro instruction may be used for all other types of remote BSC stations; however, the IBM 1800 and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

WRITE Initial Conversational Transparent (TIVX) (Using Automatic Calling List -- BSCLST)

WRITE Initial Conversational Transparent calls a remote station, writes the central computer's identification sequence and ENQ, and reads the identification sequence of the remote station and a response. If the identification matches the identification contained in the terminal list, and the response is positive (ACK-0), the operation writes message text and the ending characters DLE ETX to the remote station and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

If the identifications do not match, the Write operation ends with command (3) and is posted as complete.

1. Dial Dial digits
2. Write ID ENQ
3. Read ID ACK-0
4. Write Text
5. Write DLE ETX
6. Read Response
7. Read Text

Programming Notes:

1. This macro instruction may be used for all types of remote BSC stations. However, the IBM 1800, 2715, and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

2. This macro instruction should not be issued for an IBM 2770 or 2780 on a point-to-point line if the 2770 or 2780 requires component selection characters in the message text.[1]

If both component selection and transparent message text transmission are desired, the component selection characters should be sent in a separate message by a Write Initial operation, followed by a Write Continue (TT) or Write Continue Transparent (TTX) operation to send the transparent text.

WRITE Connect (TC) (Using Manual Calling List -- WTLIST)

A WRITE Connect macro is for use where calls to remote stations must be initiated

---

[1]Component selection characters are required unless the JOB Select switch (2770) or Mode switch (2780) is set for printing or punching (or some other output device, for 2770), in which case the message is printed or punched, regardless of the component specified by these characters.

manually by the console operator rather than by program control. Operation is as follows.

The channel program first enables the line so that calls may be initiated. After issuing the WRITE Connect macro, the program must inform the console operator (as by a WTO macro) to dial the remote station. The operator dials the call, and upon hearing a data tone from the remote station, places the data set (modem) in data mode. (This terminates the Enable command.)

The channel program then writes to the remote station the identification sequence of the central computer, then reads the identification sequence of the remote station.

If the received sequence matches the expected sequence, the operation is posted complete (without error) in the event control block. If the sequences do not match, the operation is posted complete-with-error.

This macro does not write message text to the remote station; one or more WRITE Continue macros should be issued for this purpose following the WRITE Connect macro.

1. Enable
2. Write   ID ENQ
3. Read   ID ACK-0

<u>WRITE Connect (TC) (Expanded ID Verification)(Using Automatic or Manual Calling List -- SWLST)</u>

WRITE Connect is used to originate a call to a remote BSC station, either through program-initiated (automatic) dialing or through manual dialing, and to cause exchange of identification sequences (or ENQ and ACK-0) between the central computer and the remote station.

The entry operand of the WRITE Connect macro must specify the name of a calling list of the SWLST format, as defined by a DFTRMLST macro. If the DFTRMLST macro specifies the AD operand, the automatic-dialing channel program is generated; if DFTRMLST specifies the MD operand, the manual dialing channel program is generated.

If the response from the called remote station is an ID ACK-0 sequence that matches one of the authorized ID ACK-0 sequences in the calling list, BTAM places the address of the entry containing the matching ID in the first fullword of the list and posts normal completion (X'7F').

If the response from the remote station is an ID NAK sequence, the ID portion of which matches the ID portion of one of the authorized ID ACK-0 sequences, BTAM places the address of the entry containing the matching ID in the first fullword of the list, then examines the control byte of that entry. If the control byte is 0, BTAM turns on bit 1 of DECFLAGS and posts normal completion (X'7F'). If the control byte is 1, BTAM retries the Write ID ENQ (or ENQ alone) command.

If the response from the remote station is an invalid ID sequence (i.e., one that does not match any of the authorized ID sequences in the calling list), BTAM retries the Write ID ENQ (or ENQ alone) command.

In the two foregoing situations in which BTAM retries the Write ID ENQ (or ENQ alone) command, the maximum number of retries is seven.

If all retries are unsuccessful, and a valid ID NAK sequence was received on the last retry, BTAM turns on bit 1 of DECFLAGS and posts normal completion (X'7F'). If all retries are unsuccessful, and an invalid ID sequence was received on the last retry, BTAM breaks the line connection, turns on bit 3 of DECFLAGS, and posts normal completion (X'7F').

If the response from the remote station is ACK-0 (with no preceding ID), BTAM posts normal completion (X'7F'). If the response is NAK (with no preceding ID), BTAM turns on bit 1 of DECFLAGS and posts normal completion (X'7F'). If the response is WACK, BTAM turns on bits 0 and 1 of DECFLAGS and posts normal completion (X'7F').

If no response at all is received from the remote station, BTAM retries the Write ID ENQ (or ENQ alone) command up to seven times; if all retries are unsuccessful, BTAM breaks the line connection, sets X'01' in DECSENS0, and posts a completion code of X'41'.

The channel program for automatic dialing is:

1. Dial
2. Write   ID ENQ (or ENQ alone)
3. Read   ID ACK-0 or ID NAK response

The channel program for manual dialing is:

1. Enable
2. Write   ID ENQ (or ENQ alone)
3. Read   ID ACK-0 or ID NAK response:

Programming Note: If the Write Connect operation ends with ID NAK, NAK, or WACK and you reissue the WRITE Connect macro, BTAM starts the channel program at the second command (Write ID ENQ) if the line connection is still established at the time the macro is issued. Otherwise, BTAM starts the channel program at the first command (Enable or Dial).

WRITE Continue (TT)

WRITE Continue writes message text and reads a response from the remote station.

1. Write Text
2. Read Response

Programming Notes: This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.

WRITE Continue Transparent (TTX)

WRITE Continue Transparent writes message text and the ending characters, DLE ETX, that must follow transparent data, and reads a response from the remote station.

1. Write Text
2. Write DLE ETX
3. Read Response

WRITE Continue Transparent Block (TTE)

WRITE Continue Transparent Block writes message text and the ending characters, DLE ETB, that must follow transparent data, and reads a response from the remote station.

1. Write Text
2. Write DLE ETB
3. Read Response

WRITE Continue Conversational (TTV)

WRITE Continue Conversational writes message text and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write Text
2. Read Response
3. Read Text

Programming Notes

1. This macro instruction cannot be used to send message text to an IBM 2715, because text transmission to this type of station must always be in transparent mode.

2. This macro instruction may be used for all other types of remote BSC stations; however, the IBM 1800 and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgment (ACK-0 or ACK-1).

WRITE Continue Conversational Transparent (TTVX)

WRITE Continue Conversational Transparent writes message text and the ending characters, DLE ETX, and reads a response, which may be the first two characters of a message block, an alternating acknowledgment, or NAK. If the response is message text, the remaining text is read; if not, the operation is posted complete.

1. Write Text
2. Write DLE ETX
3. Read Response
4. Read Text

Programming Note: This macro instruction may be used for all types of remote BSC stations. However, the IBM 1800, 2715, and 2770 do not transmit text as a response to text received from the central computer; they return the usual alternating acknowledgement (ACK-0 or ACK-1).

WRITE Inquiry (TQ)

WRITE Inquiry writes an ENQ and reads a response. This macro is for requesting the remote station to transmit its last response (ACK-0, ACK-1, NAK, or a conversational text reply).

1. Write ENQ
2. Read Response
3. Read Text

WRITE Wait-before-Transmit (TW)

WRITE Wait-before-Transmit writes a WACK sequence to a remote station and reads an ENQ. The purpose of this macro is to temporarily stop the remote computer from sending. You may issue it in place of READ Continue or READ Repeat, or in response to a conversational-type Write operation executed by the remote computer. The only valid responses to WACK are ENQ and EOT. You may issue Write TW repeatedly for as long as necessary to delay your regular response.

1. Write WACK
2. Read ENQ

WRITE Reset (TR)

WRITE Reset writes an EOT to indicate to the remote station that the central computer has no more message text to send, and reads a response. This macro is for giving the remote station the opportunity to transmit.

1. Write EOT
2. Read Response

WRITE Break (TB)

WRITE Break sends a Disable command to the TCU, causing the TCU to break the switched line connection. This macro does not inform the remote station that the connection is to be broken.

1. Disable

WRITE Disconnect (TD)

WRITE Disconnect writes DLE EOT, indicating to the remote station that the line connection is to be broken, then sends a Disable command to the TCU, causing the TCU to break the switched line connection.

1. Write DLE EOT
2. Disable

LOCAL IBM 3270 DISPLAY SYSTEM

For information about using READ and WRITE macro instructions for the local 3270 display system, see the section "IBM 3270 Display System - Programming Considerations."

READ MACRO INSTRUCTIONS

READ Initial (TI)

READ Initial reads modified fields from a local 3270 display station after an attention interruption has been generated by the display station operator.

1. Select command
2. Read modified command

READ Modified (TM)

READ Modified reads modified fields from a local 3270 device independently of action by the display station operator.

1. Select command
2. Read modified command

READ Modified from Position (TMP)

READ modified from Position reads modified fields from a local 3270 device beginning at a specified location in the buffer.

1. Select command
2. Write command (to set buffer address)
3. Read modified command

READ Buffer (TB)

READ Buffer reads the entire buffer of a local 3270 device.

1. Select command
2. Read buffer command

READ Buffer from Position (TBP)

READ Buffer from Position reads the entire buffer of a local 3270 device beginning at a specified location.

1. Select command
2. Write command (to set buffer address)
3. Read buffer command

WRITE MACTO INSTRUCTIONS

Write Initial (TI)

WRITE Initial writes a message to a local 3270 device.

1. Select command
2. Write command

WRITE Erase (TS)

WRITE Erase clears the buffer of a local 3270 device to nulls (binary zeros) and then writes a message to the device.

1. Erase/write command

WRITE Unportected Erase (TUS)

WRITE Unprotected Erase clears all unprotected fields in the buffer of a local 3270 device to nulls (binary zeros).

1. Erase all unprotected command
2. NOP

This chapter contains miscellaneous programming considerations for communicating between a central computer and any of the remote computers supported by BTAM as remote stations: IBM System/360 (including Model 20), IBM System/3, IBM 1130, and IBM 1800. These considerations are in addition to those shown under General Information in the BSC Read and Write Operations chapter and under the major sections within that chapter covering the three types of line configuration (nonswitched point-to-point, nonswitched multipoint, and switched point-to-point).

Except where noted, these considerations apply equally to all of the foregoing types of remote computers.

Transmission over Nonswitched Point-to-Point Line

Initial contact between the central computer and the remote computer over a nonswitched point-to-point line is on a contention basis. That is, the line remains idle until either of the computers sends an ENQ character to the other computer, signifying its intent to begin a transmission. Sending the ENQ character is called bidding for the line.

Ordinarily, both computers will not simultaneously bid for the line. On rare occasions, however, bidding will be simultaneous. When this happens, one computer must defer to the other. The control programs in the two computers must be coordinated so that this deferral takes place.

In BTAM, you accomplish this by coding MODE=CNTRL in the DCB macro for the line, if you wish the central computer (or the remote computer, if it also is running under BTAM) to retain control. Conversely, you omit MODE=CNTRL if you wish to defer to the other computer. The opposite choice must then be made for the other computer. Assume that both computers are running under BTAM. In one BTAM program, MODE= CNTRL would be specified; in the other it would be omitted. If the remote computer is running under a control program other than BTAM, the equivalent action must be taken to assure coordination between the two computers.

When this coordination is done, the computer that is to retain control automatically resends the ENQ character, and the computer that is to defer executes a Read command (or equivalent) in order to listen for the ENQ sent by the computer retaining control. Transmission begins when one computer successfully sends the ENQ character to the other; that is, receives an ACK-0 (positive acknowledgment). Thus, the direction of transmission is established by which computer sends the ENQ.

When the deferring computer is running under BTAM, and contention occurs, BTAM immediately ends the operation that sent the ENQ character and posts the operation complete-with-error (X'41' in the DECSDECB field of the DECB). When the computer retaining control is running under BTAM, and contention occurs, BTAM automatically restarts the operation to resend the ENQ character.

Transmission over Nonswitched Multipoint Line

(Not applicable for System/360 except Model 20)

Communication between the central computer and a remote computer over a nonswitched multipoint line begins when the central computer places the line in control mode by sending an EOT character, then initiates contact with the desired computer by sending that computer's polling or addressing sequence.

Transmission from remote computer to central computer is initiated when BTAM, in the central computer, executes an initial-type READ macro instruction. This causes the polling sequences for each of the remote computers to be sent automatically, in turn, until either the last computer represented in the polling list is polled (for an open-type list) or until the last computer represented in the polling list is polled following execution of a RESETPL macro for the line (for a wraparound-type list).

Transmission from central computer to remote computer is initiated when BTAM, in the central computer, executes an initial-type WRITE macro instruction. This causes the addressing sequence for the desired remote computer to be sent; the WRITE macro specifies the addressing list entry representing the desired remote computer.

The details of the polling and addressing functions are discussed under BSC Nonswitched Multipoint Operation in the BSC Read and Write Operations chapter.

The polling and addressing sequences are specified identically for all types of remote computers. That is, up to seven polling or addressing characters may be sent. There is, however, a distinction in the handling of these characters by the System/360 Model 20 and 1800 as opposed to the System/3 and the 1130. The System/360 Model 20 or 1800 hardware itself responds to the first two characters of the polling sequence. Any further characters, if any, in the sequence are merely received by the user program; the hardware does not recognize them as polling characters. For the System/3 and the 1130, however, there is no hardware-generated response to the polling or addressing sequence; all of the characters are received by the program, which decides what response to return to the central computer.

As explained under the General Information section of the BSC Read and Write Operations chapter, double addressing must be used for the System/360, System/3, and 1800; that is, the first two polling or addressing characters must be identical. Double addressing, though not required for the 1130, is advisable for the reasons mentioned in the General Information section. For the System/360 Model 20 and the 1800, the first two polling characters are set in the hardware at installation time; for the System/3 and the 1130, polling characters are specified in the program.

Transmission over Switched Point-to-Point Line

Communication between the central computer and a remote computer via the switched telephone network begins when either computer calls the other. The call may be made manually by the computer operator or it may be made automatically, where the transmission control unit at the computer is equipped with an automatic calling unit. Similarly, the operator at the called computer may answer manually or the TCU may answer automatically if it is equipped with an automatic answering unit.

BTAM provides the choice of calling a distant computer automatically or manually, and of answering calls from a distant computer automatically or manually.

Once the line connection is established, the calling computer sends an ENQ character to bid for use of the line, as is done in a nonswitched point-to-point (contention) system. The called computer, upon answering and successfully receiving the ENQ, returns an ACK-0 sequence (positive acknowledgment). Unlike a contention system, however, you may arrange for either the ENQ or the ACK-0, or both, to be preceded by from one to 15 identification characters. This allows the control program at the computer receiving the ID characters to verify that the computer sending those characters is authorized to communicate with it. In BTAM, these ID characters are user-defined in the terminal list referred to by the macro instruction that initiates the transmission. The action BTAM is to take if an invalid ID sequence or a negative acknowledgment is received may also be user-specified.

The various alternatives to be used are determined by which type of READ or WRITE macro instruction you issue to initiate the transmission and by which type of terminal list you provide for use by that macro. See the explanations in the BSC Switched Point-to-Point Operation section of the BSC Read and Write Operations chapter. Table 8 summarizes the choices.

System/3 Notes

Data Formats: IBM System/3 RPG II support uses the following formats for transmission of data. These formats must be followed when sending data to System/3 from a CPU.

- Non-transparent, non-ITB (End of Intermediate Transmission Character):

  STX-data-ETX or ETB

- Non-transparent, ITB:

  STX-data-ITB-data-ITB-data-ETX or ETB

- Transparent, non-ITB:

  DLE-STX-data-ETX or ETB

Data can be either blocked or unblocked but must be of fixed length. Fixed record length and unblocked implies non-ITB mode and requires that all data between ITBs be of the same length.

Conversational Mode: Only one response to conversational data is allowed by System/3. To maintain proper line discipline, System/3 will send or accept a NULL message (STX-ETX sequence), in lieu of a data transfer following a conversational response.

Examples of situations in which this is necessary follow:

I. System/360 Point-to-Point Contention BTAM Program (Initiate a Read)

READ TI

1. Prepare
2. Read ENQ
3. Write ACK-0
4. Read text

WRITE TT

5. Write text
6. Read response

WRITE TV

7. Write NULL message (STX-ETX sequence)
8. Read text

After the conversational transfer of data (items 4 and 5), the BTAM programmer must write a null message before continuing. If item 7 were a Write text, the System/3 would reply with an EOT.

II. System/360 Point-to-Point Contention BTAM Program (Initiate a Write)

WRITE TIV

1. Write ENQ
2. Read response
3. Write text
4. Read text

READ TT

5. Write ACK-1
6. Read NULL message (STX-ETX)

WRITE TV

7. Write text
8. Read text

After the conversational transfer of data (items 3 and 4), the BTAM programmer must realize that the System/3 will be sending a null message (item 6) instead of normal data.

WACK and TTD Responses: System/3 will transmit WACK or TTD at two second intervals during a wait time specified by an RPG programmer (default of 180 seconds). An EOT (Disc) sequence is sent after the elapsed time.

INITIATING TRANSMISSION TO AN IBM 2780

Transmission over Nonswitched Point-to-Point Line

Initial contact between the central computer and an IBM 2780 over a nonswitched point-to-point line is on a contention basis. That is, the line remains idle until either the computer or the 2780 sends an ENQ character to the other, signifying its intent to begin a transmission. Sending the ENQ character is called bidding for the line. Ordinarily, the central computer and the 2780 will not simultaneously bid for the line. On rare occasions, however, bidding will be simultaneous. When this happens, the central computer must defer to the 2780. To cause this to happen, do not code MODE=CNTRL in the DCB macro for the line, as doing so would cause BTAM to retain control.

When contention occurs, that is, both computer and 2780 send ENQ simultaneously, BTAM turns on bit 3 in the DECFLAGS field of the DECB for the line, posts the Write operation that sent the ENQ complete-with-error (X'41' in DECSDECB), and returns control to the user program. The user program should immediately issue an initial-type READ macro; this will cause BTAM to detect the next ENQ character sent by the 2780, respond by sending ACK-0, and then read message text from the 2780.

Transmission over Nonswitched Multipoint Line

Communication between the central computer and an IBM 2780 over a nonswitched multipoint line begins when the central computer places the line in control mode by sending an EOT character, then initiates contact with the 2780 by sending the addressing sequence for that 2780 or by polling the line to which the 2780 is connected.

Transmission from 2780 to central computer is initiated when BTAM executes an initial-type READ macro instruction. This causes the polling sequences for each of the remote 2780s to be sent automatically, in turn, until either the last 2780 represented in the list is polled (for an open-type list) or until the last 2780 represented in the list is polled following execution of a RESETPL macro for the line (for a wraparound list).

Transmission from central computer to 2780 is initiated when BTAM executes an initial-type WRITE macro instruction. This causes the addressing sequence for the desired 2780 to be sent; the WRITE macro specifies the addressing list entry representing the 2780.

The details of the polling and adressing functions are discussed under BSC Nonswitched Multipoint Operation in the BSC Read and Write Operations chapter.

The polling sequence for an IBM 2780 is always

x 6 ENQ

where the x may be any uppercase or lowercase character used as the station address of the 2780. This address is established at the time the 2780 is installed. The second character is always a 6 to indicate the card reader.

The addressing sequence for an IBM 2780 is always

x 3 ENQ (to address the printer) or

x 4 ENQ (to address the punch)

where the x is the station address, as described for the polling sequence.

Polling and addressing examples are shown in Figure 22.

Transmission over Switched Point-to-Point Line

Communication between the central computer and an IBM 2780 via the switched telephone network begins when either the computer or the 2780 calls the other. The call may be made manually by the computer or terminal operator or it may be made automatically, where the 2780 or the transmission control unit at the computer is equipped with an automatic calling unit. Similarly, the operator at the computer or the 2780 may answer manually or the computer or 2780 may answer automatically if it is equipped with an automatic answering unit.

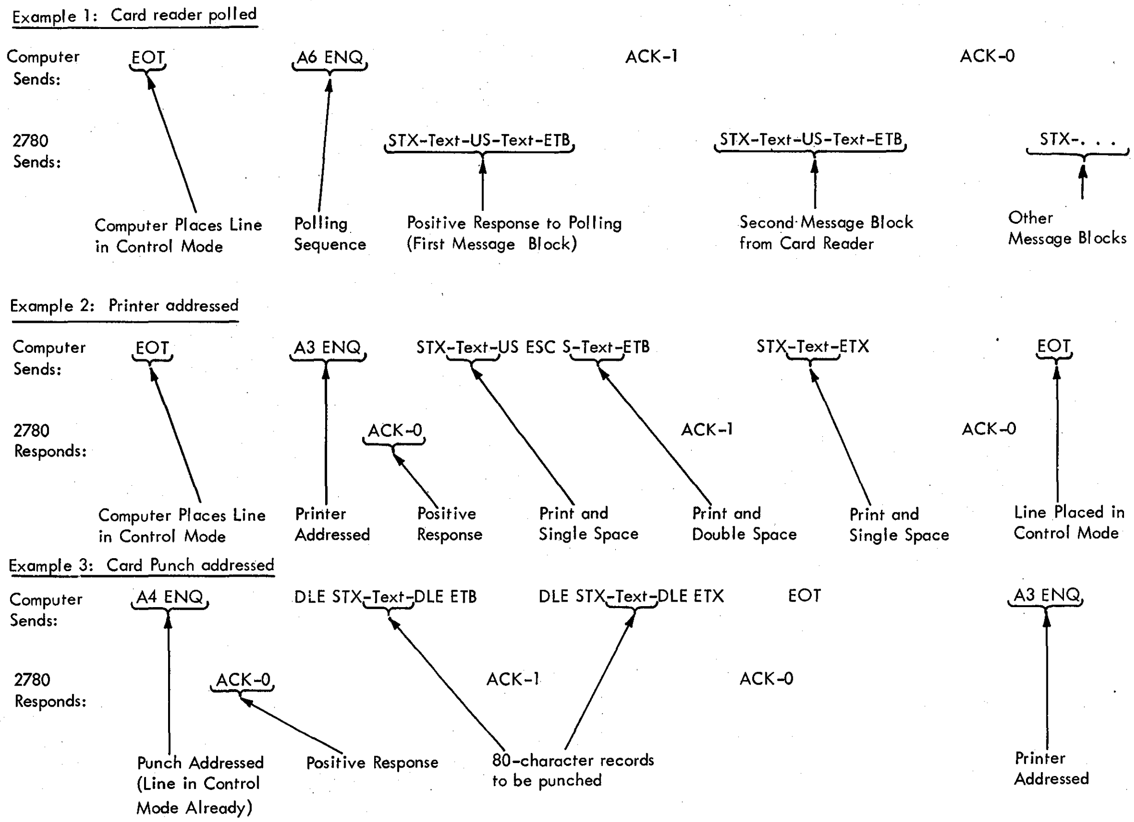


Figure 22. Multipoint Polling/Addressing Operations for 2780

Once the line connection is established, the calling station (computer or 2780) sends an ENQ character to bid for use of the line, as is done in a nonswitched point-to-point (contention) system. The called station, upon answering and successfully receiving the ENQ, returns an ACK-0 sequence (positive acknowledgment). Unlike a contention system, however, you may arrange for either the ENQ or the ACK-0, or both, to be preceded by from one to 15 identification characters. This allows the 2780 operator or BTAM to verify that the station sending those characters is authorized to communicate with it. In the 2780 these characters, like the station address, are established at the time the 2780 is installed. In BTAM, these characters are user-defined in the terminal list referred to by the macro instruction that initiates the transmission. The action BTAM is to take if an invalid ID sequence or a negative acknowledgment is received may also be user-specified.

The various alternatives to be used are determined by which type of READ or WRITE macro instruction you issue to initiate the transmission and by which type of terminal list you provide for use by that macro. See the explanations in the BSC Switched Point-to-Point Operation section of the BSC Read and Write Operations chapter. Table 8 summarizes the choices.

END-TO-END CONTROL CHARACTERS

In the character set of the IBM 2780 are four characters that provide secondary end-to-end control functions required by the printer, card reader, and card punch of the 2780.

One of these characters, BEL, is not used when the 2780 communicates with the central computer. (Its function in terminal-to-terminal operation is to cause the audible alarm to sound.)

Another end-to-end control character is EM (end-of-medium). When the card reader detects this character punched in a card it is currently reading, the reader ejects the card. When the EM character is sent to the card punch, it causes the punch to eject the card. These functions occur, however, only when the EM appears in nontransparent text. If it appears in transparent text, it is ignored. Regardless of whether it is in nontransparent or transparent text, however, it is punched in the card, when sent to the card punch.

The remaining two end-to-end characters are ESC (Escape) and HT (Horizontal Tab). The ESC character is used in component selection (except multipoint lines), in vertical forms control, and in horizontal formatting. The HT character is used in horizontal formatting, when the 2780 is equipped with the Printer Horizontal Format Control feature.

The use of the ESC and HT characters is discussed under the functions with which they are associated.

COMPONENT SELECTION

In communicating with an IBM 2780 over a switched or nonswitched point-to-point line (but not a multipoint line), you must perform component selection when transmitting messages to the 2780, if the Mode switch at the 2780 is set at Transmit or Receive. If the Mode switch is set to Print or Punch, the 2780 ignores component selection messages; all messages it receives are automatically printed or punched. The Mode switch is manually set by the 2780 operator.

Component selection is accomplished by sending a nontransparent message beginning with a two-character escape sequence. To select the punch, begin the message (following the STX character) with ESC 4. To select the printer, begin the message (after STX) with any one of the vertical forms control escape sequences (e.g., ESC /, ESC A, ESC B). Thus, the vertical forms control escape sequences (discussed below) perform the dual function of selecting the printer and controlling the forms motion for the records in which they appear.

Once a component is selected, you need not reselect it in successive records to be sent to that component.

Because the 2780 recognizes component selection escape sequences only in nontransparent messages it receives, the first message, and any subsequent messages containing component selection sequences, must always be in nontransparent mode. Intervening messages can be in transparent mode. Figure 23 illustrates the use of component selection sequences.

Nonswitched Point-to-Point Line

When the 2780 finishes sending messages to the central computer, it transmits ETX (or DLE ETX). The computer responds to this with a positive acknowledgment (ACK-0 or ACK-1) if it detected no errors. Upon receiving the positive acknowledgment, the 2780 sends an EOT to the computer.

When the 2780 is equipped with the Automatic Turnaround feature, the card punch automatically enters ready status after the card reader reads a blank card. This permits the computer, upon receiving EOT, to immediately bid for the line (by sending ENQ) and select the punch. Without this feature, selection of the punch following a card reader operation requires operator intervention. Automatic selection of the printer, however, does not require the presence of the Automatic Turnaround feature.

Switched Point-to-Point Line

The 2780 normally does not transmit the disconnect signal (DLE EOT) to the central computer; it thus lets the user program at the central computer decide when to break the switched line connection after a transmission. As mentioned above, when the 2780 receives a positive acknowledgment to the last message it sends, it returns an EOT to the computer. The user program may then send the disconnect signal (DLE EOT) or, via a WRITE Inquiry macro, bid for use of the line.

When the user program sends an EOT to the 2780, the 2780 reacts in one of two ways.

If the card reader is in ready status, the 2780 bids for use of the line by sending ENQ. Upon receiving the ENQ, BTAM posts normal completion (X'7F' in the DECSDECB field of the DECB). The user program may, if it is ready to receive, then issue a READ Continue (TT or TTL) macro.

If the card reader is not in ready status, the 2780 does not respond to the EOT. Instead, it continues to monitor the line for an ENQ from the computer. If it receives no ENQ within about 20 seconds, the 2780, if equipped with the Automatic Answering feature, will then break the line connection (go "on-hook").

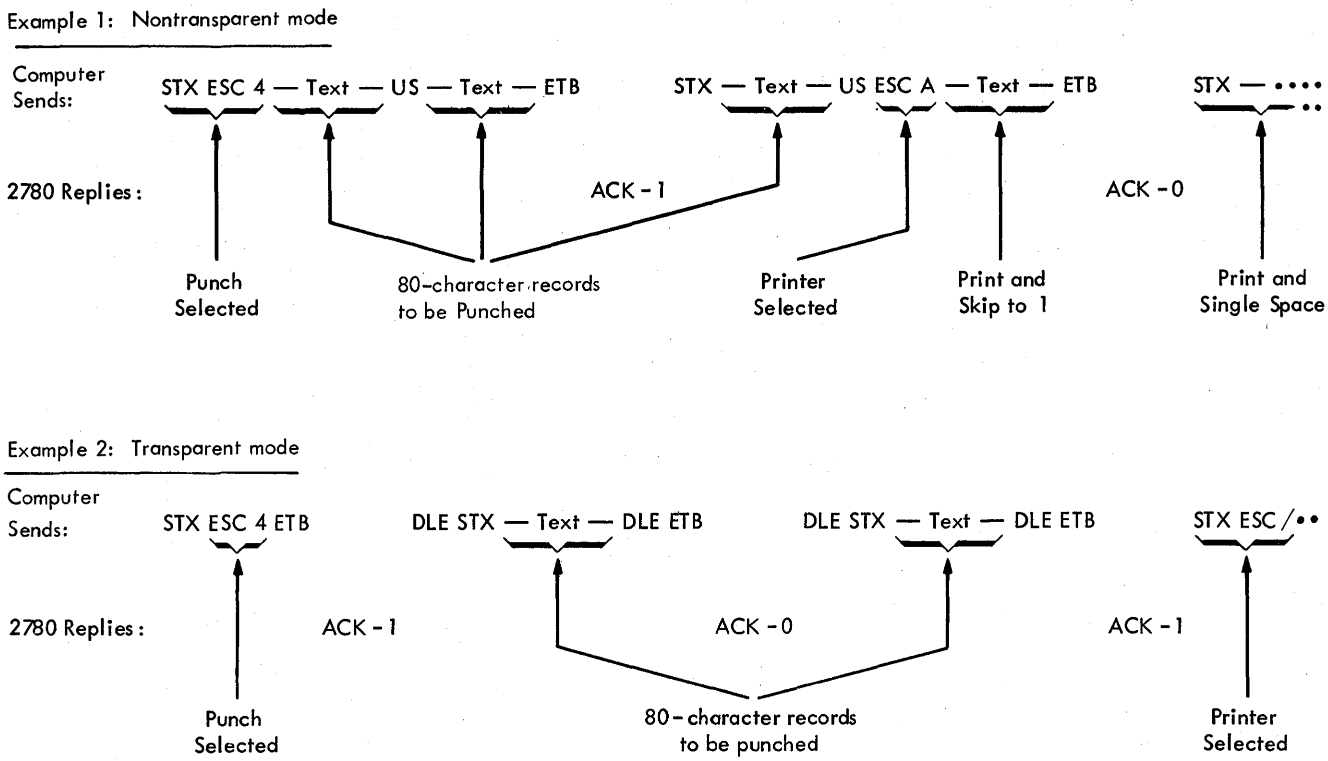


Figure 23. Examples of Component Selection for 2780

Vertical Forms Control

Vertical positioning of forms at the 2780 printer is controlled by predefined escape sequences. These are shown in Figure 24.

You must place the sequence in the first two character positions of each message block (print line record) you send to the printer. Exception: If you wish single spacing and, for a switched or nonswitched point-to-point line, you require no component selection characters, you may omit the escape sequence (ESC /, for EBCDIC, or ESC Q, for USASCII). For double or triple spacing, or skipping to a channel of the printer carriage control tape, you must begin the print line record with the appropriate escape sequence. See Example 2 of Figure 22 for examples of escape sequences for vertical forms control.

Escape Sequence

USASCII	EBCDIC, Transcode	Forms Motion After Printing
ESC Q	ESC /	Single space
ESC R	ESC S	Double space
ESC S	ESC T	Triple space
ESC A	ESC A	Skip to ch. 1
ESC B	ESC B	Skip to ch. 2
ESC C	ESC C	Skip to ch. 3
ESC D	ESC D	Skip to ch. 4
ESC E	ESC E	Skip to ch. 5
ESC F	ESC F	Skip to ch. 6
ESC G	ESC G	Skip to ch. 7
ESC H	ESC H	Skip to ch. 8

(In EBCDIC, the ESC character is identical to the PRE (Prefix) character.)

Figure 24. Vertical Forms Control Escape Sequences

HORIZONTAL FORMAT CONTROL

The IBM 2780 may be equipped with the Printer Horizontal Formatting Control feature. This feature allows the printer of the 2780 to receive from the line instructions on arranging across the page the text contained in subsequent print line records it receives. To tell the 2780 the format desired, the user program at the central computer must send a format record each time the format is to be changed. The format record most recently received by the printer governs the horizontal format of all subsequent print line records. (Sending a format record is equivalent to setting the tab stops on a typewriter.)

Each format record must begin with the ESC HT sequence. The record contains other HT characters, and intervening SP (space) characters such that a HT appears at each position of the format record corresponding to the print line position where a stop is to be made (i.e., a tab stop is to be set), with SP characters occupying all other format record positions. No characters other than HT and SP may appear in the format record. It is not necessary to make the format record as long as the print line; it may end at the last HT character. The final character of the format record is an ETB. See Figure 25 for an example.

Format records must always be sent in nontransparent mode.

The 2780 printer is governed by the last-received format record until:

- it receives a new format record
- power is removed from the terminal
- a card is read by the card reader (EBCDIC and USASCII codes only)
- a record is received by the card punch (EBCDIC and USASCII codes only)

For terminals using Transcode, the printer retains the format record even if card reading or punching occurs following use of the printer.

Once a format record has been sent to the printer, HT characters are used in subsequent print line records the user program sends to the printer to cause skipping to the next stop position on the print line. This is equivalent to pressing the Tab key of a typewriter.

<u>Example:</u> Assuming that the printer is governed by the format record shown in Figure 25, if you wished to print the two characters A,B at print positions 1 and 2, and the three characters C,D,E at positions 12, 13, and 14, you would send this print line record:

STX A B HT HT C D E ETB

If HT characters appear in nontransparent text when no format record is in effect, or if it appears in the text beyond the format record position corresponding to the last stop on the print line, an error occurs--overrun of the print line.

If HT characters appear in transparent text, the 2780 does not recognize them as end-to-end control characters but treats them as text characters.

If the 2780 is not equipped with the Printer Horizontal Formatting Control feature, the 2780 does not recognize HT characters as end-to-end control characters regardless of whether they appear in transparent or nontransparent text.

STX $\underbrace{\text{ESC HT}}_{1}$ $\underbrace{\text{SP SP SP SP}}_{2}$ $\underbrace{\text{HT}}_{3}$ $\underbrace{\text{SP SP SP SP SP SP}}_{4}$ $\underbrace{\text{HT}}_{5}$ $\underbrace{\text{SP SP SP}}_{6}$ $\underbrace{\text{HT}}_{7}$ ETB

1. Two-character sequence introducing the format record.
2. Indicates print positions 1-4 do not contain a stop.
3. Indicates a stop at print position 5.
4. Indicates no stop in print positions 6-11.
5. Indicates a stop at print position 12.
6. Indicates no stop in print positions 13-15.
7. Indicates a stop at print position 16.

Figure 25. Example of a 2780 Format Record

<u>MULTIPLE RECORD TRANSMISSION</u>

The 2780 contains a 400-character buffer for receiving data from the communications line. This buffer can accommodate one, two, or more records depending on whether the terminal is equipped with the Multiple Record Transmission feature and whether transparent or nontransparent mode is used.

<u>2780 without Multiple Record Transmission Feature</u>

Without the feature, the 2780 can receive or send two nontransparent records per transmission. The first record is ended by the unit separator (US) character (also called interchange unit separator -- IUS -- in EBCDIC) which provides the ITB function, while the second record ends with ETB or ETX. Thus, in nontransparent mode, a single transmission apppears on the line as:

STX ...text... US ... text... ETB (or ETX)

When received by the 2780, only the text is printed or punched. The US and ETB (or ETX) cause the punch to eject the card, but are not themselves punched into the card (unlike EM (end-of-medium), which is punched into the card).

The STX character is mandatory at the beginning of the first record, and optional at the beginning of the second (after US).

Upon receiving a two-record transmission, the 2780 checks each record individually. If the first record is valid and the second is invalid, it prints or punches only the first record. If the central computer subsequently resends the two-record transmission, the 2780 prints or punches only the second record, as the first has already been successfully printed or punched. If in the initial transmission the first record is invalid, the 2780 prints or punches neither record, even if the second record is valid. This avoids processing records out of sequence.

The 2780 can send two transparent records per transmission, in this format:

DLE STX ...text... DLE US DLE STX ... text... DLE ETB (or DLE ETX)

(The distinction between ETB and ETX is that ETX is used to end the last block in a transmission.)

Although the 2780 can send two-record transmissions in transparent mode, BTAM can send the 2780 only one record per transmission. The format is:

DLE STX ...text... DLE ETB (or DLE ETX)

2780 with Multiple Record Transmission Feature

With the feature, the 2780 can send or receive up to seven records per transmission, in nontransparent mode, and can send up to four records, in transparent mode. BTAM can send the 2780 only one record per transmission in transparent mode.

The maximum number of records per transmission, and the maximum number of data characters per record, are shown in Figure 26.

Although escape sequences cannot perform component selection in transparent mode, they may be used in transparent-mode transmissions to the printer to perform vertical forms control. They are optional in this case, however, and do not count as part of the fixed record length shown in Figure 26. The only case in which the escape sequence is mandatory in transparent mode is if the first data character in the record coincides with the ESC character.

Mode	Maximum number of records per transmission	Maximum number of data characters[1] per record
Nontransparent	2 without MRT 7 with MRT	(variable length record) 80 (to punch) 80 (from reader) print line[2] (to printer)
Transparent	1 (to 2780) 2 (from 2780 w/o MRT) 4 (from 2780 with MRT)	(fixed length record) 80 (to punch) 80 (from reader) print line[2] (to printer)

[1]Not including control characters (STX, US, etc.) or escape sequences.
[2]The print line may be 80, 120, or 144 positions.

Figure 26. IBM 2780 Records: Number and Length

PROGRAMMING CONSIDERATIONS

Maximum Transmission Length

The maximum number of characters in one transmission to the 2780 is 400. All characters in the transmission, including escape sequences, end-to-end control characters, US, ETB, and ETX, occupy positions in the buffer, with one exception. STX characters are deleted by the 2780 control unit as they arrive and do not appear in the buffer. The limitation of 400 is in addition to other restrictions already mentioned, such as print or punch record length and number of records per transmission.

Automatic Turnaround Feature

If the 2780 is equipped with the Automatic Turnaround feature, the first blank card placed in the card reader hopper behind cards being read and transmitted will cause the 2780 to stop card reading and send an ETX to the central computer. (The blank card is not sent over the line). In this case the last message block from the 2780 will be:

STX ...text... US ETX

Nonmixing of Modes within a Transmission

Within one transmission you cannot send records in both transparent and nontransparent modes. That is, a transmission of the form

STX ...text... US DLE STX ...text... DLE ETB

is not permitted.

Responses by 2780 to Abnormal Conditions

There are two possible responses by the 2780 to abnormal conditions occurring when the 2780 is receiving a transmission from the central computer: NAK and EOT.

The 2780 sends a response of NAK for line errors. For example, the received message block may contain a parity error. The 2780 returns a NAK to request the central computer to retransmit. Another kind of line error may result in in a received message block containing too many characters. (A user-program error could produce the same result.) Again, the 2780 returns a NAK to request retransmission. Clearly, if the user program has actually sent too many records or characters in one transmission, the user program must be corrected; no amount of retransmission could clear the error condition.

The 2780 sends EOT in response to certain user-program errors that it can detect, such as failure to send a format record to the printer before sending print line records containing horizontal format control, or sending too many characters in a record.

The 2780 also sends EOT when it detects certain internal errors, such as a punch jam, printer forms check, buffer parity check, etc. When these conditions occur, the 2780 immediately responds to the central computer with EOT, signifying that the 2780 operator must intervene before operation can resume.

Recovery from such a disruption depends on the user-program restart and recovery procedures and on the 2780 operator procedures. The operator's instructions may, for example, tell him to prepare the 2780 to expect retransmission of the message block to which the 2780 responded with EOT. Or he may be instructed to prepare the 2780 to receive a new message block, unrelated to the previous one. In any event, the operator's response and the user program's response to these conditions should be consistent.

In certain instances the 2780 will not immediately respond with EOT when an internal error occurs. This happens only if the error condition occurs during the printing or punching of the last record in a received message block. Upon verifying that the last record in a block is free of error, the 2780 responds with the appropriate acknowledgment to the central computer and overlaps processing of the last-received record with receipt of the next block. If an internal error occurs during processing of the last record, the 2780 responds to the <u>next</u> block with an EOT. The 2780 operator can intervene to process the record on which the error actually occurred without retransmission of that record. The user program would then retransmit the next message block.

A description of internal errors and operator procedures may be found in the publication <u>IBM 2780 Data Transmission Terminal Component Description</u>, GA27-3005.

In sending to the central computer, the 2780 makes special use of the ENQ character to indicate the occurrence of an internal error. If a buffer parity or overrun error occurred in the message block it is sending, the 2780 sends a message block of the form:

```
STX ...text...  US ....text...  ENQ
   (or)
STX ...text...  ENQ
```

A message ending with ENQ is invalid, and BTAM error recovery procedures automatically respond to the 2780 with a NAK character. The 2780 then sends EOT to indicate that it is aborting transmission until the operator corrects the error condition.

If an internal error occurs during processing of a record being readied for transmission to the central computer, it sends STX ENQ (with no text). The central computer and 2780 then exchange NAK and EOT, as above.

When it receives the EOT, BTAM turns on bit 7 of the DECFLAGS field of the DECB and posts abnormal completion, with a code of X'41'. Subsequent transmission by the 2780, after communication with the central computer is resumed, normally begins with the record on which the error occurred.

BTAM supports Binary Synchronous Communication between a System/360 and an IBM 2790 Data Communications System (via an IBM 2715 Transmission Control Unit) over point-to-point (switched and nonswitched) and multipoint nonswitched line configurations.

The 2715 is available in two models: local and remote. The local 2715 (2715 Model 1) is attached directly to the multiplexer channel of a System/360. A Binary Synchronous Module is provided in the 2715 Model 1 to simulate a Binary Synchronous Communications Adapter, operating on a point-to-point nonswitched line. The remote 2715 (2715 Model 2) can be connected to an IBM 2701 Data Adapter Unit attached to a System/360 multiplexer or selector channel, or to an IBM 2703 Transmission Control attached to a System/360 multiplexer channel. Communication between the System/360 and the 2715 (local and remote) follows the line control conventions of Binary Synchronous Communications. To the System/360 programmer, a local 2715 is indistinguishable from a remote 2715 (operating on a point-to-point nonswitched line).

For detailed descriptions of the 2790 system, see the <u>IBM 2790 Component Description</u> manual, GA27-3015.

The transmission code supported is transparent EBCDIC. When communicating with a System/360, the 2715 sends error information and diagnostic messages to the system. The type of message (error information or diagnostic) is indicated in a special header that precedes the message text and is transparent to the user.

- Diagnostic messages are operator awareness messages that are printed on the System/360 console or the 2740, if available. These messages are the result of the 2715 completing diagnostics.

- Error information messages are recorded on disk. These messages are sent when the 2715 has filled a sector of its integral disk with error information data or by user request.

<u>Note</u>: If one of these diagnostic or error information messages is received by BTAM on a READ Initial operation, BTAM changes the optype to READ Continue and restarts the operation. The initial flag bit in the DECB is left on in this case. If one of these messages is received on a multipoint READ Initial operation, the input area may contain the index byte and EOT.

The READ, WRITE, and CONTROL options available for the 2715 are listed in Table 6.

BTAM provides a set of macro instructions to enable the user to describe the processing that must be performed on his input. The 2715 cannot be programmed by the user. IBM-supplied microcode in the 2715 interprets processing requirements through a set of tables generated by the user-coded macros. There are 18 user macros that can be used with the 2790 System. Seven of the macros are for the basic system four are for the Pulse Count feature, and seven are for the 2798 Guidance Display Unit. The 18 macros, when assembled by the System/360 Assembler, generate 21 types of tables. The tables are transmitted in object form to the 2715 by the users BTAM program. These tables contain pointers and index values, as well as parameters used by the microcoded routines. (See Appendix M for a sample 2790 program.)

2715 PULSE COUNT FEATURE

The 2790 Data Communications System is a data collection and data communication system. The 2715 Pulse Count feature is provided to allow the 2790 system user to dynamically control and monitor production work flow. This feature adds another major capability to the 2790 system.

Some of the highlights of the 2715 Pulse Count feature are:

- The 2793 Area Station is the only area station in the 2790 system on which pulse counters can be attached.

- Up to 63 counters are allowed on a 2793 area station.

- Up to 1008 counters are allowed on a 2790 system.

- These counters can have a decimal count from 0 to 29,999.

- The READ/WRITE capability is available for all counters.

- The overflow interrupt capability is available on all counters.

- Implicit/explicit counter addressing at the DEU level is permitted.

- Count testing can be performed on up to 504 counters in the system.

- Schedule readout capability is available for up to 504 counters in the system.

The 2715 Pulse Count feature has many possible uses. Some of the functions that can be performed with it are:

- Appending counts automatically to transactions entered on a 2795/2796/2797 Data Entry Unit.

- Monitoring the current progress of counters by requesting readouts of counters for printing at the 2740, the area station 1053 printer, or the System/360.

- Setting counters to predetermined counts, and when these counts are reached, automatic printouts of the counters are routed to the 2740, the area station 1053, or the System/360.

- Monitoring for unassigned production with printout notification at the 2740, the area station 1053, or the System/360.

- Monitoring counters on a scheduled basis and informing the user when a counter is not advancing. Printouts can be directed to the 2740, area station 1053, or the System/360 to alert the user of such "no-count" production conditions.

- Readout of counters on a scheduled basis for analysis by users.

- Scheduling up to 15 user-selected time schedules for flexibility in implementing the "count test" and "scheduled readout" functions described above.

Counter Testing

Two types of counter testing can be automatically performed by the 2715 if the user so desires. The user specifies one of 15 possible test schedules for each of 504 counters by means of the user table. He can also specify the type of count testing that is to be automatically performed on each of up to 504 counters. The 2715 will scan the counter table and perform one of two count tests, "no count" or "unassigned production," depending on what the user specifies in the tables and also whether or not count testing is enabled.

The user has the ability to enable or disable count testing from either the System/360 or the 2790 DEUs via user-specified transaction lists. When initiated from a DEU, the desired action must be specified in the transaction list (CTRLIST macro). If explicit counter addressing is specified, this address must be within the data entry. Implicit counter addressing at a DEU implies counters 1 through 32 only. All count test entries must be the last data entry from the DEU.

The user can enable or disable count testing from the System/360 or the 2715 operator's console for all counters on an area station, or all counters on the system. If the user disables count testing for all counters on the system, he can enable all counters and either have the 2715 continue from the previous stop point in the timing of the schedules, or have the 2715 re-initialize all schedules and start again.

Three possible count testing actions can be initiated on an individual counter basis:

1. No-count test can be started. The 2715 automatically stops unassigned production testing in this case.

2. Unassigned production test can be started. The 2715 automatically stops no-count test in this case.

3. All count testing can be stopped.

The no-count test informs the user that a counter is not advancing. The no-count test is executed on a user-defined schedule on an individual counter basis. When the 2715 detects that a counter is not advancing, a message signifying a no-count condition is generated and routed to the destination defined by the user in the ROUTE operand of the ASCTR macro instruction. The message indicates which counter has not advanced when it should have. The 2715 then disables further no-count testing for that counter until the user has corrected the situation and has enabled further testing. Count testing for all other counters remains in progress.

The unassigned production test informs the user that a counter is advancing when it should not. The unassigned production test is executed on a user-defined schedule on an individual counter basis. When the 2715 detects a counter advancing, a message signifying an unassigned production test condition is generated and routed to the destination designated by the user, in the ROUTE operand of the ASCTR macro instruction. This message indicates which counter has advanced when it should not. The 2715 then disables further unassigned production testing for that counter until the user has corrected the situation and has enabled further testing. Count testing for all other counters remains in progress.

Scheduled Counter Readout

An individual counter readout function is provided so that the 2715 will automatically read up to 504 counters on user-defined schedules if the user so desires. The user specifies one of 15 possible schedules for each of up to 504 counters in the table macros (CTRGROUP and CTRSCHED). As the 2715 reads a counter, the transaction header is attached and the transaction is stored automatically on disk as deferred data. The transaction header contains the counter address and associated time stamp.

The user can enable or disable scheduled readout from the System/360 or the 2715 operator's console for all counters on an area station or all counters on the system. He can also enable or disable scheduled readout for a single counter from the System/360. If the user disables scheduled readout for all counters on the system, he can enable scheduled readout on all counters and either have the 2715 continue from the previous stop point in the timing of the schedules or have the 2715 re-initialize all schedules and start again.

DEU Set Counter Capability

The initiating transaction from a DEU consists of one or more data entries. For multiple data entries, the set counter function is contained within the last data entry. There is no set counter capability on the 2795 DEU. Only the 2796 and 2797 DEUs have this capability. The user selects the proper transaction list by setting the top left knob on the 2796 or the left knob on the 2797. (Transaction expansion may not be used.) The last step in the transaction list must be the counter appendage. The counter control byte in the counter appendage step of the transaction list specifies the set counter operation and also specifies whether or not implicit or explicit counter addressing is to be used. For implicit addressing, the user does not have to specify the counter address. The DEU address is automatically translated to a counter address. The user specifies the high-order byte of the 5-digit value to which the counter is to be set by turning the top right knob on the 2796 or the right knob on the 2797 to the high-order digit of the value. This digit must be 0,1, or 2, because the value cannot exceed 29,999. The user specifies the low-order four digits of the counter value in the four digit-rocker switches on the 2796 or by keying them in on the manual entry digit keys of the 2797.

Note: There are 6 possible manual entry positions on the 2797. Only the 4 low-order positions are used for the low-order value of the set counter operation.

Explicit addressing requires that the counter address be contained in the data entry. Explicit addressing may be accomplished by DEU manual entry or by manual entry and card or badge entry. If manual entry is used, the lower left and lower right knobs on the 2796 or the two high order positions of the 2797 manual entry field specify the counter address. If manual and card or badge entry are used, columns 77 and 78 of the card or columns 19 and 20 of the badge must contain the counter address.

The status of a DEU-initiated set counter operation is indicated via normal status reporting. Unsuccessful set counter transactions initiated by a DEU are signified by raising the red error flag on the DEU. Set counter functions may not be routed to ASLOG printer.

DEU Read Counter Capability

The initiating transaction from the DEU consists of one or more data entries. For a transaction that contains multiple data entries, the last data entry must contain the information necessary to initiate a read counter. The 2795, 2796, and 2797 DEUs have the read counter capability. The user selects the proper transaction list by setting the left knob on the 2795 or 2797 or the top left knob on the 2796. The last step in the transaction list must be the counter appendage. The counter control byte in the counter appendage step of the transaction list specifies the read counter operation to be performed, and also specifies whether implicit or explicit counter addressing is to be used. Read counter values are appended to the normal transaction and routed according to the user-defined routing designation in the transaction list.

The read counter capability includes both implicit and explicit counter addressing for all read operations except Read Group and Read Group Residual, for which explicit counter addressing must be used. The following read operations may be defined in the last step of the transaction list:

- Read (single or group)
- Read Residual (single or group)
- Read and Reset (single counter)
- Read and Set* (single counter)

*For 2796 and 2797 only.

For implicit counter addressing, the user does not have to specify the counter address. The DEU address is automatically translated to the counter address. For explicit counter addressing, the user may manually set the lower-left and lower-right knobs on the 2796 to the counter address, or he may put the counter address in columns 77 and 78 of the card or columns 19 and 20 of the badge. For the 2797, the user may manually enter the two digits for the counter address in the manual entry digit keys (these two digits must be left justified), or he may put the counter address in columns 77 and 78 of the card or columns 19 and 20 of the badge. Explicit counter addressing for the 2795 may only be specified from columns 77 and 78 of the card or columns 19 and 20 of the badge. Manual entry is not possible on the 2795.

For explicit counter addressing on the Read Group or Read Group Residual, the user may manually set the lower-left and lower-right knobs on the 2796 to the starting counter address, and the first two digit-rocker switches to the ending counter address, or he may put the starting and ending counter addresses in columns 77-80 of a card or 19-22 of a badge. For the 2797, the user may manually enter the two digits for the starting counter address followed by the two digits for the ending counter address in the manual entry digit keys (these four digits must be left justified), or he may put the starting and ending counter addresses in columns 77-80 of a card or columns 19 through 22 of a badge. For the 2795, the user must specify the starting and ending counter addresses in columns 77-80 of a card or columns 19 through 22 of a badge. For a Read and Set operation, the user specifies the set counter value in the same way as for the set counter operation (transaction expansion may not be used), with the top right knob and the four digit-rocker switches on the 2796, or the right knob and the four low-order digits from the manual entry digit keys on the 2797.

The counter transaction and count value may be logged at any area station for 1053 display by message routing, but the counter transaction must consist of only one step in addition to the counter appendage. Any of the read operations (except Read Group and Read Group Residual) may be routed to an area station 1053 printer by implicitly or explicitly specifying the output destination. Read and Set operations (READSET) may not be routed to the ASLOG printer. The implicit routing address is in the transaction list, while the explicit routing address is from the card or badge entry. Implicit message routing is done by using both the left and right knobs on the 2795 and 2797 or the upper left and right knobs on the 2796 to address the transaction list. For explicit message routing to an area station 1053, the user must specify the 1053 address in columns 71 and 72 of the card or columns 13 and 14 of the badge. This applies to any DEU.

Note: Transaction expansion is a prerequisite to message routing. Storage expansion (32K) is a prerequisite to transaction expansion.

Setting a counter to a certain value implies that the user wants to know when the counter reaches that value. The set counter function sets a counter to a value of 29,999 minus the value specified, so that when the user-specified count is reached, the user is alerted to an overflow interrupt for that counter. The read counter function allows the actual value of the counter to be read. The read residual counter function allows the value of the counter to be subtracted from 29,999, so that the residual difference is read.

For example, if a user sets a counter to a value of 10 and immediately performs a read counter function, the value read will be 29,989. If he immediately performs a read residual function on the counter, the value read will be 10.

2715 TABLES

The types of tables are:

- Table Definition Block.
- Area Station Table.
- Data Entry Unit Table.
- Transaction Group Tables.
- Transaction List Tables.
- Area Station Sequence Table.
- Area Station Counter Table.

- Counter Table.
- Schedule Table.
- Data Entry Unit Sequence Table.
- Data Entry Unit Index Table.
- System Parameter Table.
- Transaction Table.
- GDU List Table.
- Parameter List Number Table.
- Parameter List Table.
- Display Guidance Table.
- GDU Area Station Table.
- GDU Sequence Table.
- Identification Table.
- Translate Table.

Descriptions of these tables and the macros that generate them follow.

Note: The user must assemble all his macros at the same time since the relationship among the tables is established by labels.

Table Definition Block: The Table Definition Block contains a pointer to each of the other tables. It is defined by the CONFIGUR macro instruction.

Area Station Table (AS Table): The Area Station Table contains one entry per area station. Each entry is one byte and contains a numeric pointer that relates the specified area station to a particular transaction group within the Transaction Group Table. The maximum size of the AS Table is 100 bytes for a 2715 having 32K bytes of storage and 64 bytes for a 2715 having 16K bytes of storage. The AS Table is defined by the AS macro instruction.

Data Entry Unit Table (DEU Table): The Data Entry Unit Table contains one entry for each area station defined in the system. This entry (0-99) is used for all data entry units attached to the designated area station. If there are no attached data entry units, the entry contains a value to indicate this condition. The DEU table is defined by the AS macro instruction.

The position of the entry in the table is relative to the position of the area station address within the valid range of addresses. For example, the first entry in the DEU Table is for the data entry units attached to the area station with ID=0; the second for those attached to the area station with ID=1; etc.

Each entry in the DEU Table is one byte and is used to gain access to the transaction group associated with all the data entry units attached to the area station. This indicates that all data entry units attached to an area station must use a common transaction group.

Transaction Group Table (TGROUP Table): Each transaction group consists of nine halfword (two-byte) entries that contain pointers to a transaction list or to another transaction group. Each entry corresponds to a transaction code (a transaction key on an area station or the value of the left rotary knob on a 2795 or 2797 or the top left rotary knob on a 2796 Data Entry Unit). Each entry contains a pointer to a transaction list that defines the operating procedure associated with the specified transaction code. If nine transaction lists are not sufficient, an indication can be set in one or more of the transaction group entries to permit a transaction expansion function in which a secondary value (the first digit of input from an area station or the value of the right hand rotary knob on a data entry unit) is used to index another transaction group. Therefore, it is possible for an area station or data entry unit to refer to nine TGROUP entries, any or all of which may indicate secondary indexing. This allows a data entry unit to perform a maximum of 81 distinct transaction functions, while allowing 81 functions for area stations (see TGROUP in the Macro Descriptions section).

All area stations that have the same operating characteristics must refer to the same transaction group, using the area station address and the corresponding entry in the AS Table. The same is true for data entry units, using the area station address and the corresponding entry in the DEU Table. There can be up to 63 transaction groups, each of which uniquely specifies an area station or data entry unit capability. The transaction groups are defined by TGROUP macro instructions.

Transaction List Tables (TRLIST Tables): Each Transaction List Table consists of a three-byte identification and routing header field and either an internal message or from one to sixteen data entry steps. The header field determines the destination of the completed transaction. Each data entry step is generated by an ASLIST or DEULIST macro and determines whether checking is to be performed on the input. If an ASLIST macro generated a data entry step, the step contains the number of the next guidance light to be turned on (more than one guidance light number is included if the user chooses to include error checking in his transaction step, for example, via LENGTH and DIGIT operands of the ASLIST macro).

A TRLIST Table is defined in any one of three ways:

- A TRLIST macro followed by one or more ASLIST macro instructions,
- A TRLIST macro followed by one or more DEULIST macro instructions,
- A TRLIST macro followed by one or more ASLIST (DEULIST) macro instructions with specification for message routing with an internal message.

The first Transaction List Table always refers to all IBM 1035 Badge Readers, if there are any on the system.

Area Station Sequence Table (AS-SEQ Table): A transaction from an area station may comprise a discrete number of processing steps (for example, badge, card, card). The AS-SEQ Table keeps track of the last step of the transaction entered from each area station. The AS-SEQ Table has one entry per area station. Each entry is one byte and contains the step number (0-15). The maximum size of the AS-SEQ Table is 100 bytes, one byte for each of the 100 possible area stations. The AS-SEQ Table is defined by the AS macro instruction.

Area Station Counter Table (ASCTR Table): The Area Station Counter Table contains one entry per area station. Each entry is two bytes and contains a displacement to the group of counters in the Counter Table for that particular area station. Each entry also contains routing information for counter overflow and count test response messages. All counters attached to an area station have counter overflow and count test response messages routed to the same destination.

Each entry in the Area Station Counter Table is used to gain access to the counters in the Counter Table associated with this area station. The displacement in each entry, plus the counter address, allows the 2715 to index to individual counters. Scheduled readout and count testing are performed at the individual counter level.

The maximum size of the Area Station Counter Table is 202 bytes: 2 bytes for each of 100 area stations, plus 2 additional control bytes. This table is truncated at the highest assigned area station. Unassigned area stations below the highest assigned area station require 2 bytes of unused main storage. However, the highest assigned area station with counters can be any assigned area station less than or equal to the highest assigned area station defined by the AS macro operand ID=n. The Area Station Counter Table is defined by the ASCTR macro instruction.

Counter Table (CTR Table): The Counter Table consists of 2 bytes of control information for each counter in the system that requires testing. This table is organized on a group basis. Each area station that has one or more counters (to be tested) constitutes a group. Each group is truncated at the highest counter address.

Counters for which scheduled readout and count testing are not to be done require 2 bytes each in the counter table if their addresses are less than the highest counter address. Only 504 counters can be defined in the Counter Table. The Counter Table is defined by CTRGROUP macro instructions. Each CTRGROUP macro defines a single counter.

Schedule Table: The Schedule Table consists of 2 bytes for each user-defined count test schedule or readout schedule. The maximum number of schedules that can be defined is 15, so the maximum size of the Schedule Table is 30 bytes. These schedules determine the frequency (in minutes) with which the Counter Table entry will be tested. Scheduled readout and count test operations can refer to any of the 15 schedules that are defined. The maximum frequency that can be specified is 2047.

System Parameter Table: The System Parameter Table has a fixed core location and maintains an index of other 2798 GDU table locations. The System Parameter Table contains the following information:

System error guidance:
  Invalid function error guidance
  Premature termination error guidance
  Monitor key error guidance
Table pointers:
  GDU Sequence Table pointer
  Transaction Table pointer
  Parameter List Number Table pointer
  Identification Table pointer
  Translate Table pointer
  GDU AS Table pointer

Identification characters:
  "Get" identification character
  "Store" identification character

Counts:
  GDU list count
  Identifier count

The System Parameter Table is generated by the CONFIGUR macro. The information in the table is determined from the CONFIGUR macro operands. There is only one System Parameter Table per 2790 System with 2798 GDUs.

Transaction Table: The Transaction Table is used to index the desired GDU list from a GDU operator entered transaction code. Each entry in the Transaction Table may contain a pointer to a GDU list associated with a transaction code. There are 100 possible transaction codes so there are 100 possible entries in the Transaction Table. Each entry in the table is 2 bytes long for a maximum table size of 200 bytes. The table is truncated at the highest assigned transaction code. But for each entry skipped between zero and the highest assigned value two bytes of core are reserved, just as if this value had been assigned. Each entry in the Transaction Table is generated by a GDUTRANS macro.

GDU List Table: The GDU List Table contains up to 100 GDU lists. Each GDU list entry will contain a transaction header, a variable number of GDU steps (up to 16), an all zero byte, and implicit text (if specified). Each GDU list entry is generated by a combination of the TRLIST macro and from 1 to 16 GDULIST macros. The TRLIST macro generates the transaction header. Each GDULIST macro generates a 5 byte GDU step with the following information

- A one-byte parameter list number.
- Two bytes of normal guidance to be sent to the operator guidance panel on the GDU. This guidance is used to light a combination of 16 lights.
- A two-byte display guidance pointer used to:
  a. Point to a display guidance message in the Display Guidance Table, or
  b. Point to an identifier in the Identifier Table

Implicit text is defined in the last GDULIST macro in the GDU list entry. Each GDU list entry can contain a minimum of 10 bytes and a maximum of 85 bytes plus implicit text.

Parameter List Number Table: The Parameter List Number Table contains up to 127 addresses of the parameter lists. Each entry in the table is two bytes for a maximum table size of 254 bytes. This table is generated by PARAMNUM macros, each of which generates a two byte entry. The table is truncated at the highest defined parameter list number defined by the PLN operand of the PARAMNUM macro.

Parameter List Table: The Parameter List Table contains up to 127 entries. The data in each parameter list entry defines the types of checks that are performed on a data entry. The first two bytes of a parameter list contain a check field and a function field in which the checks and/or functions associated with this list are denoted. Following these two bytes are the check lists if any are required. These check lists provide the test information and error guidance for the checks performed on a data entry. The check lists are variable lengths, depending on the tests to be done. The parameter lists are packed decimal. Each parameter list is generated by a PARMLIST macro.

Display Guidance Table: The Display Guidance Table contains the various messages used as display guidance for the GDUs on the loop. Each entry in the Display Guidance Table contains a length byte and from 1 to 16 data bytes. Each entry in the table is generated by a DISPGUID macro.

Guidance Display Unit Area Station Table (GDUAS Table): The Guidance Display Unit Area Station Table, in conjunction with the GDU device address, contains pointers which provide entries into the GDU Sequence Table. Each entry is two bytes and is generated by the GDUAS macro. The maximum size of the GDUAS table is 200 bytes. This table is truncated at the highest assigned area station. Unassigned area stations below the highest assigned area station require two bytes of unused main storage.

Guidance Display Unit Sequence Table: The Guidance Display Unit Sequence Table consists of one byte of zeroes for each GDU on the 2790 system. Each byte is used for inquiry display and GDU sequence. This table is generated by the GDUTRANS macro.

Identification Table: The Identification Table contains the areas to maintain the GDU identifiers. Fourteen bytes are generated for each identifier. The total number of identifiers is specified in the System Parameter Table. This table is generated by the STEND macro.

Translate Table: The Translate Table is used in conjunction with the translate function and contains a maximum of eight entries. Each entry contains a translate character, the length of the text into which it is to be translated, and from 1 to 14 bytes of text. One byte of zeroes follows the last entry in the table. The maximum table size is 129 bytes. Each entry in this table is generated by a TRANSLAT macro.

Data Entry Unit Sequence Table (DEU-SEQ Table): A transaction from a data entry unit may comprise a discrete number of steps. The DEU-SEQ Table keeps track of the last step entered from each data entry unit on the system. Each one-byte entry contains the step number (0-15) of the transaction initiated by each data entry unit. The maximum size of the DEU-SEQ Table is 1,024 bytes (one byte for each of the 1,024 data entry units available on the 2790 system). The DEU-SEQ Table is defined by the AS macro instruction.

Data Entry Unit Index Table (DEU-INDEX Table): When a data entry is received from a data entry unit, the 2715 receives the area station address and the data entry unit address. The area station address is used as an index to the desired entry in the DEU-INDEX Table. (There is one entry per area station.) The one-byte entry in the DEU-INDEX Table contains a pointer to the desired entry in the DEU-SEQ Table. However, since the DEU-SEQ Table can be up to 1,024 bytes long, two additional bits are necessary. Using bits 6 and 7 of the corresponding entry in the DEU Table extends the addressing capability of the DEU-INDEX Table entry to the full 1,024 possible entries of the DEU-SEQ Table. Use of the data entry unit address allows indexing to the specific counter associated with the data entry unit sending the transaction. The DEU-INDEX Table is defined by the AS macro instruction.

Loading the Tables

The assembly output of the user macros is a set of tables, in object form, needed for the 2715 internal operation. The user must write a BTAM routine to load these tables into the 2715. During transmission of the object text from the System/360 to the 2715, the user must include two headers before each card's data: a message header and a transaction header (Figure 28). The user must provide DLE STX in front of the message (see format below). BTAM provides the ending characters. All cards must be transmitted, one at a time, to the 2715.

D L E	S T X	Message Header 2 bytes	Transaction Header 8 bytes	Object Card 80 bytes

When loading the tables, the user must first be sure that the 2790 system is inactive. This is accomplished by using the defined control transactions to "stop 2790 input" and to purge the disk of all deferred data ("Read deferred data"). When activity at the 2715 has ceased, the user must initially send a "table load start" control transaction. This is followed by the transmission of the table object cards as "table load data" control transactions (see the Message Format section).

Columns 73-80 of the object deck contain a program identification and a sequence number, which are checked by the 2715. The program identification (columns 73-76) is determined from a named TITLE card generated by the CONFIGUR macro instruction. Both the identification and the sequence are checked by microcoded routines in the 2715. The completion of the data load is signaled by the END card (END in columns 2-4). After transmitting the END card, the user must send a "table load end" control transaction. If an error is found in either the program identification or the sequence field of any card, the table load is rejected.

When transmitting the tables, the maximum message length is 128 bytes. When the last message of the table load has been sent, the user should transmit an EOT. The 2715 bids for the line and then transmits a message indicating the status of the table load (see the Message Format section). The table load is rejected if any of the following conditions occurs:

1. Invalid program identification sequence field.
2. Improper control field in transaction header.
3. 2715 table size exceeds storage available.
4. Data has not been purged from the 2715 integral disk.
5. The system is active.

MACRO INSTRUCTIONS

The macros coded for the 2790 System must be in the following order:

- CONFIGUR
- AS
- GDUAS (optional)
- TGROUP
- ASCTR (optional)
- CTRGROUP (optional)
- CTRSCHED (optional)
- GDUTRANS (optional)
- PARAMNUM (optional)
- PARMLIST (optional)
- DISPGUID (optional)
- TRANSLAT (optional)
- TRLIST
- ASLIST
- DEULIST (optional)
- CTRLIST (optional)
- GDULIST (optional)
- STEND

Configuration Macro (CONFIGUR): The Configuration macro generates the table definition block that contains pointers to the other user tables.

Area Station Definition Macro (AS): The Area Station Definition macro permits building an exhaustive list of all area stations present in the system. In addition, each macro logically attaches area stations and data entry units to their associated transaction groups.

Guidance Display Unit Area Station Macro (GDUAS): The GDUAS macro is used to build an entry in the GDU Area Station Table.

Transaction Group Macro (TGROUP): By coding the Transaction Group macro, the user establishes a pointer to a set of transaction lists that can be associated with the transaction keys of a group of area stations, with the left-hand knob positions of a group of 2795 or 2797 Data Entry Units, or with the top left-hand knob positions of a group of 2796 Data Entry Units.

With each of the nine transaction keys on an area station, the operator can select up to nine transaction lists. With each position of the left-hand knob on a 2795 or 2797 or of the top left-hand knob on the 2796 Data Entry Unit, the operator can also select up to nine transaction lists (position 0 is reserved). Normally, each transaction key or position of the left-hand knob is associated with only one transaction list. Since groups of area stations and groups of data entry units usually have the same operating procedure, a given set of area stations must have corresponding transaction keys associated with identical transaction lists.

Area Station Counter Macro (ASCTR): By coding the ASCTR macro, the user defines each area station that has pulse counters, and establishes a displacement to the counter group in the Counter Table associated with each particular area station. In addition, the user defines routing information for counter overflow and count test response messages for all the counters on each area station.

Counter Group Macro (CTRGROUP): By coding the CTRGROUP macro, the user can define two bytes of control information for each counter on an area station for which scheduled readout or count testing is to be done.

Counter Schedule Macro (CTRSCHED): The CTRSCHED macro defines the count test schedules and the readout schedules that can be used by all the counters on the system.

Guidance Display Unit Transaction Macro (GDUTRANS): The GDUTRANS macro is used to build an entry in the Transaction Table. A GDUTRANS macro must be coded for every transaction code that will be used in communication with the 2798 GDUs.

Parameter List Number Macro (PARAMNUM): The PARAMNUM macro is used to define an entry in the Parameter List Number Table for use with 2798 GDUs.

Parameter List Macro (PARMLIST): The PARMLIST macro is used to generate an entry in the Parameter List Table. The data in this entry defines the types of checks that are performed on a data entry from a 2798 GDU.

Display Guidance Macro (DISPGUID): The DISPGUID macro is used to define a display guidance message in the Display Guidance Table.

Translate Table Macro (TRANSLAT): The TRANSLAT macro is used to build an entry in the Translate Table.

Transaction List Macro (TRLIST): The Transaction List macro is used with the Area Station List macro or the Data Entry Unit List macro to define a transaction, or it is used to define a user-specified message. The TRLIST macro enables the user to control the destination of the completed transaction or a predefined message. Each transaction list has a DEU or AS transaction code associated with it.

Area Station List Macro (ASLIST): When the transaction is to be initiated by an area station, the Transaction List macro is followed by one or more Area Station List macros. The ASLIST macro is used to define one step of a transaction for a 2791 Area Station; the transaction code is the value of the transaction key pressed by the operator.

Data Entry Unit List Macro (DEULIST): When the transaction is to be initiated by a data entry unit, the Transaction List macro is followed by one or more Data Entry Unit List macros. The DEULIST macro is used to define one step of a transaction for a data entry unit; the transaction code is the value of the left-hand rotary knob of a 2795 or 2797 and of the top left-hand rotary knob of a 2796.

Guidance Display Unit List Macro (GDULIST): The GDULIST macro is used to define one step of a GDU transaction list for a 2791 or 2793 Area Station with 2798 GDUs attached.

Counter List Macro (CTRLIST): When the transaction is to be initiated by a data entry unit, the TRLIST macro is followed by one or more DEULIST macros, and, optionally, by a CTRLIST macro instruction. The CTRLIST macro defines the last step of a transaction for a data entry unit that is attached to a 2793 Area Station using pulse counters.

Statement End Macro (STEND): The Statement End macro indicates the end of all user macros.

Macro Descriptions

The macros are arranged in the following section in the same order as they must appear in the assembly.

The macros must be assembled together. The first assembly statement must be the CONFIGUR macro (there must not be a TITLE, CSECT, or START card). The last assembly statement must be the END card. There must not be any other macros or code inserted into the assembly of the user tables.

CONFIGUR (Configuration) Macro Instruction

The CONFIGUR macro is used to generate the table definition block, which contains pointers to the 2715 tables. The format of the CONFIGUR macro is:

Name	Operation	Operand
[symbol]	CONFIGUR	[CORE={<u>16</u>|32}] [,PC={<u>NO</u>|YES}] [,GDU={<u>NO</u>|YES}] [,FUNCERR=(absexp,...)] [,ENDERR=(absexp,...)] [,MONERR=(absexp,...)] [,GETID=absexp] [,STORID=absexp] [,IDCOUNT=absexp] [,INQDISP={<u>NO</u>|YES}]

symbol
The name of the macro is optional.

CORE
The CORE parameter specifies the 2715 storage size. The only valid values are 16 and 32. If an invalid value is specified, an MNOTE is issued and no code is generated. If the CORE parameter is omitted, 16 is assumed.

If CORE=16 is coded, then the size of all tables built must not exceed 1,280 bytes. If CORE=32 is coded, then the size of all tables built must not exceed 4,096 bytes. The size of all tables built is calculated by using Table 8.1. The size will be the total of all macros used.

PC
The PC operand indicates whether pulse count macros are coded in this assembly. If PC=YES is specified, pointers to the Area Station Counter Table, the Counter Table, and the Schedule Table are included in the Table Definition Block. Coding PC=YES adds six bytes to the Table Definition Block. If PC=NO is coded, the pointers to the Area Station Counter Table, the Counter Table, and the Schedule Table are omitted, but two bytes of zeros are added. If the PC operand is omitted, PC=NO is assumed.

The PC operand is valid only if CORE=32 is specified. If PC=YES is coded and CORE=16 (or the CORE operand is omitted), an MNOTE is issued and no code is generated.

Table 8.1 2715 Macro Storage Size Estimates

MACRO	MIN. BYTES USED	ADDITIONAL CONSIDERATIONS
CONFIGUR	22	Add 4 if PC=YES. Add 22 If GDU=YES. Add 14 times the number coded if IDCOUNT=n.
AS	4	Add 4 for each skipped ID in sequence. Add 1 for each DEU attached.
GDUAS	1	Add 1 for each skipped ID in sequence.
TGROUP	18	
ASCTR	2	Add 2 for each skipped ID in sequence. Add 2 after last ASCTR macro.
CTRGROUP	2	Add 2 for each skipped counter in sequence.
CTRSCHED	2	For each schedule.
GDUTRANS	2	Add 2 for each skipped TRCODE operand.
PARMLIST	2	Add 3 for CKLNGTH=. Add 3 for CKMOD11=. Add 3 for CKMOD10=. Add 8 for CKOR=. Add 3 for CKNONUM=. Add 3 for CKNUM=. Add 5 for CKRANGE= plus 1 for each position in CKRANGE field. Add 3 for CKAND= plus 1 for each position in CKAND field.
DISPGUID	1	Add 1 for each text character.
TRANSLAT	16	
TRLIST	5	
ASLIST	5	Add 1 for each implicit text character.
DEULIST	5	Add 1 for each implicit text character.
GDULIST	5	Add 1 for each implicit text character.
CTRLIST	5	Add 1 for each implicit text character.
STEND	0	Add 1 for each area station in system if INQDISP=YES in CONFIGUR macro.

FUNCERR
This operand indicates the error guidance that will be returned to the 2798 GDU when the following invalid functions are recognized:

- An invalid length is specified on a Get Identifier function. Normally, the GDU operator keys the 'Get ID' character, the two digit identifier address, and the ENTER key upon entering a GDU step having a get condition in its function field. An invalid length occurs when more than two characters are entered for the identifier address folling the 'Get ID' character.

- An invalid length on a Translate function. This error occurs when a GDU step is entered that has a translate function associated with it and more than one keyed char-

- An invalid address on a Get Identifier or Store Identifier function. This error occurs when the two digit identifier address is not in the Identifier Table, or the two digit identifier address has not been specified.

- A non-translatable character is specified on a Translate function. This error occurs when the character to be translated is not found in the Translate Table.

- A non-numeric character is recognized during a range check.

The value of each suboperand in this operand may range from 1 to 16 and up to 16 suboperands may be coded. This operand indicates exactly which lights on the Guidance Display Panel the user wants to turn on when the error occurs. (See Figure 27.1).

ENDERR
: This operand indicates the error guidance that will be returned to the 2798 GDU when a premature termination occurs. A premature termination occurs when the number of characters received in a data entry from a 2798 is not sufficient to complete all of the checks specified by this GDULIST macro (with exception of the CKLENGTH check specified by the PARMLIST macro). The value of each suboperand in this operand may range from 1 to 16 and up to 16 suboperands may be coded. This operand indicates exactly which lghts on the Guidance Display Panel the user wants to turn on when the error occurs. (See Figure 27.1).

MONERR
: This operand indicates the error guidance that will be returned to the 2798 GDU when a Monitor key check error occurs. The value of each suboperand in this operand may range from 1 to 16 and up to 16 suboperands may be coded. This operand indicates exactly which lights on the Guidance Display Panel the user wants to turn on when the error occurs. (See Figure 27.1)

GETID
: This operand specifies the GET Identification character that is used for the GET Identifier function. The value of this operand is the hexadecimal equivalent of any of the characters: A-Z, 0-9, and any of the special characters: $ & - / , # @ " = : ? ! ; * + TAB NEWLINE LINEFEED SPACE. (See Table 9.1)

STORID
: This operand specifies the Store identification character that is used for the Store Identifier function. The value of this operand is the hexadecimal equivalent of any of the characters: A-Z, 0-9, and any of the special characters: . $ & - / , # @ " = : ? ! ; * + TAB NEWLINE SPACE LINEFEED. (See Table 9.1)

IDCOUNT
: This operand specifies the number of identifiers that will be used. The value of this operand may be from 0 to 100. This operand must be coded if GDU=YES. Everytime the 2715 is ICPLed, the predefined text 'NOT USED' will be defined in every identifier in the identifier table. The user should use the Store Identifier function prior to using the Get Identifier function after an ICPL if he expects useful information to be in the identifier table.

INQDISP
: This operand indicates whether Inquiry Display will be used on the 2715. If this operand is omitted, INQDISP=NO is assumed. If INQDISP=YES is coded, CORE=32 must also be coded.

AS (Area Station) Macro Instruction

The AS macro is used to build an entry in the Area Station Table and the Data Entry Unit Table, and a corresponding entry in the Data Entry Unit Index Table. In each table, the position of the entry to be built relative to the beginning of the table is determined from the ID parameter. Each AS macro requires (4 + absexp) bytes of 2715 storage (absexp is the value of the DEGROUP operand). The format of the AS macro is:

Name	Operation	Operand
[symbol]	AS	ID=absexp[,ASGROUP=symbol] [,DEGROUP=(symbol,absexp)]

symbol
: **The name field is optional.**

ID
: **Each area station is assigned a specific address (X'80' - X'E3') at system installation time, and the value of "absexp" is the decimal representation of that address (see Table 9).** ID is used to determine the position

of an entry in the AS Table, DEU Table, and DEU-INDEX table. The value of the ID parameter must be from 0 to 99 when CORE=32 in the CONFIGUR macro; however, all values in this range need not be specified. The value of the ID parameter must be from 0 to 63 when CORE=16 or when the CORE operand is omitted in the CONFIGUR macro; however, all values in this range need not be specified. If one is omitted, a warning message is generated. The values 100 through 128, which would generate hexadecimal values E4 through FF, are not valid. The AS macros must be in ascending sequence by ID. An AS macro found to be out of sequence or in error terminates the assembly of this macro instruction. The ID of a macro in error is subsequently handled the same as an omitted ID.

ASGROUP
The ASGROUP parameter is valid for the 2791 only. The value of "symbol" is the name of the transaction group for this area station and must appear in the name field of a TGROUP macro. If the name does not appear, an assembly error occurs. The ASGROUP parameter builds an entry in the AS table.

DEGROUP
symbol
The name of the transaction group with which the data entry units on this area station are associated is specified by "symbol". It must appear in the name field of a TGROUP macro; if not, an assembly error occurs. "symbol" builds an entry in the DEU Table. "symbol" must be identical to the name of the first (or only) TGROUP macro that defines a transaction group for data entry units. That is, "symbol" must be the same as the name of the transaction group for the 2795s connected to this area station, or of the dummy transaction group that precedes the transaction group for the 2796s connected to this area station (see Figure 27 for examples).

absexp
The value of the absolute expression is the number of data entry units attached to this area station. This number must not exceed 32, since this is the maximum number of data entry units that can be attached to any one area station. The value is used to build an entry in the DEU-INDEX table.

The DEGROUP operand may be omitted if no data entry units are attached to this area station. The entries in the DEU table and the DEU-INDEX table corresponding to this area station are then defined with a value indicating there are no DEUs. The DEGROUP parameter must be coded for the 2793.

GDUAS (Guidance Display Unit Area Station) Macro Instruction

The GDU Area Station macro instruction is used to build an entry in the GDU Area Station Table. The position of the entry to be built relative to the beginning of the table is determined from the ID operand. Each GDUAS macro requires 2 bytes of 2715 storage. The format of the GDUAS macro is:

Name	Operation	Operand
[symbol]	GDUAS	ID=absexp,GDUNUMB=absexp

symbol
The name field of this macro is optional.

ID
Each area station with attached 2798 GDUs is assigned a specific address (X'80' - X'E3') at system installation time, and the value of 'absexp' is the decimal representation of that address (see Table 9). ID is used to determine the position of an entry in the GDUAS Table. The value of the ID operand must be from 0 to 99. The GDUAS macros must be in ascending sequence by ID. A GDUAS macro found to be out of sequence or in error terminates the assembly of this macro instruction.

GDUNUMB
This operand specifies the number of GDUs on this area station. The value of this operand can be from 1 to 16 for a 2793 and from 1 to 12 for a 2791 Model 3.

Note: The GDUAS macro should only be coded for an Area Station with 2798 GDUs attached. The GDUAS macro must be coded once for every area station with 2798 GDUs attached.

TGROUP (Transaction Group) Macro Instruction

The TGROUP (Transaction Group) macro defines entries in a TGROUP Table. Each keyword operand associates a transaction list with a transaction code. The maximum number of TGROUP macros allowed is 63. The TGROUP macro instructions must follow the last AS macro instruction coded. A macro sequence error occurs if they do not. Each TGROUP macro requires 18 bytes of 2715 storage.

The three types of data entry units (2795, 2796, and 2797) require separate transaction groups, one for all 2795s connected to an area station, followed by one for all 2796s connected to the same area station, and immediately followed by one for all 2797s connected to the same area station.

If all three types of DEUs are connected to the same area station, the user must code three TGROUP macro instructions. The first defines the transaction group for the 2795s, the second defines the transaction group for the 2796s, and the third defines

Value of ID Parameter in AS or GDUAS Macro (Decimal)	Valid Addresses of Area Stations (Hexadecimal)	Value of ID Parameter in AS or GDUAS Macro (Decimal)	Valid Addresses of Area Stations (Hexadecimal)	Value of ID Parameter in AS or GDUAS Macro (Decimal)	Valid Addresses of Area Stations (Hexadecimal)
0	80	33	A1	66	C2
1	81	34	A2	67	C3
2	82	35	A3	68	C4
3	83	36	A4	69	C5
4	84	37	A5	70	C6
5	85	38	A6	71	C7
8	86	39	A7	72	C8
7	87	40	A8	73	C9
8	88	41	A9	74	CA
9	89	42	AA	75	CB
10	8A	43	AB	76	CC
11	8B	44	AC	77	CD
12	8C	45	AD	78	CE
13	8D	46	AE	79	CF
14	8E	47	AF	80	D0
15	8F	48	B0	81	D1
16	90	49	B1	82	D2
17	91	50	B2	83	D3
18	92	51	B3	84	D4
19	93	52	B4	85	D5
20	94	53	B5	86	D6
21	95	54	B6	87	D7
22	96	55	B7	88	D8
23	97	56	B8	89	D9
24	98	57	B9	90	DA
25	99	58	BA	91	DB
26	9A	59	BB	92	DC
27	9B	60	BC	93	DD
28	9C	61	BD	94	DE
29	9D	62	BE	95	DF
30	9E	63	BF	96	E0
31	9F	64	C0	97	E1
32	A0	65	C1	98	E2
				99	E3

Table 9. AS Macro ID Parameter Decimal and Hexadecimal Equivalents

the transaction group for the 2797s. If only 2797s are connected to the area station, three TGROUP macros must still be coded. The first two define dummy transaction groups and the third defines the transaction group for the 2797s. If only 2796s are connected to the area station, two TGROUP macros must be coded. The first will be a dummy and the second will define the transaction group for the 2796s. If only 2795s are connected to the area station, one TGROUP macro must be coded. The dummy groups are required because the 2715 microcoded routines expect to find the groups for the three types of DEUs in the same relative main storage position from the beginning of the user tables.

Where two or more TGROUP macros are coded for the same area station, the name of the first macro must appear in the DEGROUP operand of the AS macro for that area station. The first operand of any dummy TGROUP macro must be identical to the first operand of the following non-dummy TGROUP macro for the same area station. Only one operand need be coded for dummy TGROUP macros.

Figure 27 shows examples of how TGROUP macros may be coded.

The format of the TGROUP macro is:

Name	Operation	Operand
symbol	TGROUP	[TCn=(symboln[,E])]

symbol
The name field is required for this macro instruction.

TCn
If TCn=symboln is coded, the transaction code "n" is associated with the transaction list referred to by "symboln". At least one TCn operand must be present. An MNOTE is issued if all operands are omitted. The value of "n" must be between 1 and 9 inclusive. If TCn=(symboln,E) is coded, it indicates that this is a transaction expansion entry and that the transaction is associated with the transaction group referred to by "symboln". The transaction group referred to by "symboln" must have transaction code 1 defined. When using the transaction expansion function, the format of the TGROUP referred to varies for area stations and data entry units. The first character of data received indicates the desired entry in the TGROUP Table.

```
Name      Operation  Operands
* DEFINING TRANSACTION GROUPS FOR BOTH 2795 AND 2796 DATA ENTRY UNITS

          AS         ID=59,DEGROUP=(GROUP1,5) (TOTAL OF 5 DEU'S)
GROUP1    TGROUP     TC1=ATTENDNC,TC2=SETUP,TC3=PRODN,... (2795'S)
GROUP2    TGROUP     TC1=MESSAGE,TC2=RECEIPT,TC3=ISSUE,... (2796'S)

* DEFINING TRANSACTION GROUP FOR 2796 AND DUMMY
  TRANSACTION GROUP FOR 2795

          AS         ID=59,DEGROUP=(GROUP1,2)  (TOTAL OF 2 2796'S)
GROUP1    TGROUP     TC1=MESSAGE     (DUMMY TABLE FOR 2795'S)
GROUP3    TGROUP     TC1=MESSAGE,TC2=RECEIPT,TC3=ISSUE,...  (2796'S)

* DEFINING TRANSACTION GROUP FOR 2795 ONLY

          AS         ID=59,DEGROUP=(GROUP1,3)  (TOTAL OF 3 2795'S)
GROUP1    TGROUP     TC1=ATTENDNC,TC2=SETUP,TC3=PRODN,...  (2795'S)

* DEFINING TRANSACTION GROUP FOR 2797 AND DUMMY TRANSACTION
  GROUP FOR 2795 AND 2796

          AS         ID=59,DEGROUP=(GROUP1,3)  (TOTAL OF 3 2797'S)
GROUP1    TGROUP     TC1=MESSAGE  (DUMMY TABLE FOR 2795'S)
GROUP2    TGROUP     TC1=MESSAGE  'DUMMY TABLE FOR 2796'S)
GROUP3    TGROUP     TC1=MESSAGE,TC2=RECEIPT,TC3=ISSUE,...(2797'S)
```

Figure 27. Examples of Defining Transaction Groups For Data Entry Units

Each entry in the transaction group referred to points to a transaction list. If the transaction expansion is for an area station, the first step of each of these transaction lists must be identical. Transaction expansion must be used when generating the transaction list in which message routing is to be specified. A transaction expansion entry must not refer to another transaction expansion entry.

ASCTR (Area Station Counter) Macro Instruction

The ASCTR macro is used to generate the Area Station Counter Table. The Area Station Counter Table requires two bytes of control information for each of up to 100 area stations, plus two additional bytes, for a maximum of 202 bytes. This table is truncated at the highest assigned area station, that is, the area station with the highest ID. Unassigned area stations below the highest assigned area station will each have two bytes defined in the Area Station Counter Table by BTAM at assembly time. The highest assigned area station with counters can be any assigned area station less than or equal to the highest assigned area station that is defined by the AS macro operand ID=n.

Associated with each area station with counters is a displacement that provides for scanning of the counter table. This displacement is used with the counter address to provide an index to individual counter level control. Schedule readout and count testing are performed at the individual counter level.

This macro also allows routing specification to be specified for counter overflow and count test response messages. All counters attached to an area station must have these messages routed to the same destination.

The format of the ASCTR macro instruction is:

Name	Operation	Operand
[symbol]	ASCTR	ID=absexp, HIGHCTR=absexp, ROUTE=({CPU | DISK} [,LOG] [,ASLOG] [,EXTALRM]) [,NEXTAS=absexp]

The macro is coded once for each area station with pulse counters. The maximum number of ASCTR macros that can be coded is 100.

symbol
: The name field is optional.

ID
: specifies the decimal representation of the address of the area station on which pulse counters are available. The ID operand is not required for all area stations, but the ID operands must be in ascending sequence. An ID operand out of sequence causes an invalid table assembly. The ID operand may have values from 0 to 99.

HIGHCTR
: specifies the number of the highest counter on this area station that scheduled readout or count testing may be performed on. Values for the HIGHCTR operand may range from 0 to 63, since only 63 counters are allowed on any given area station. A value of 0 indicates that no area station counters use scheduled readout or count testing.

ROUTE
: specifies the destination of counter overflow and count test response messages. At least one destination must be specified, and if only one is specified, the parenthesis are not coded. The CPU and DISK suboperands are mutually exclusive.

CPU
: specifies that counter overflow and count test response messages should be routed directly to the CPU.

DISK
: specifies that counter overflow and count test response messages should be routed to the 2715 integral disk.

LOG
: specifies that counter overflow and count test response messages should be routed to the 2740 attached to the 2715.

ASLOG
: specifies that counter overflow and count test response messages should be routed to the area station 1053 printer from which the overflow was initiated.

EXTALRM
: specifies that counter overflow and count test response messages should be routed to the 1053 printer on the area station from which the overflow was

initiated, and that the external alarm contact closure at the area station should be activated.

The above suboperands specifying routing information for counter overflow and count test response messages need not be coded in any given order.

NEXTAS
: specifies the decimal representation of the address of the next higher area station with pulse counters on which count testing or readout functions may be scheduled. The NEXTAS operand may have values from 0 to 99. This operand must be coded when HIGHCTR=0 is coded, but is not necessary for any other HIGHCTR value. NEXTAS=0 must be coded if there is no higher area station that has pulse counters on which count testing or readout functions may be scheduled. Unless NEXTAS=0 is coded, the NEXTAS operand must be greater than the ID operand for this ASCTR macro. If the NEXTAS operand is greater than 0, the ASCTR macro referred to by the value of the NEXTAS operand must have a HIGHCTR operand value greater than 0.

CTRGROUP (Counter Group) Macro Instruction

The CTRGROUP macro is used to generate the Counter Table. The macro must be coded once for each counter in the system on which schedule readout or count test functions are to be performed. A CTRGROUP macro must be coded for the counter whose value was specified in the HIGHCTR operand of the ASCTR macro instruction for this area station. The counter Table is organized on a group basis. Each group consists of the highest counter with scheduled readout or testing and all counters (whether scheduled or unscheduled) below it on the same area station. Each group is truncated at the highest counter scheduled for readout or testing, with a maximum of 63 counters allowed per area station. The Counter Table consists of two bytes of control information for each of these counters (scheduled and unscheduled) plus two additional bytes at the end of the table. Up to 504 counters may be scheduled for the entire system. Each CTRGROUP macro defines two bytes of control information for a particular counter.

The format of the CTRGROUP macro instruction is:

Name	Operation	Operand
[symbol]	CTRGROUP	ctrno, [sro], [cttest], ID=absexp [,SROENAB={NO (default) | YES}] [,CTINIT={NULL (default) | NCT | UNASP}]

symbol
: The name field is optional.

ctrno
: identifies the counter on which schedule readout or count testing is to be done. The value of the ctrno operand must be from 1 to 63; however, all values in this range need not be specified. All counters must be specified in ascending sequence.

sro
: indicates which readout schedule is to be used for this particular counter. The value of the sro operand must be from 0 to 15. A 0 value indicates that schedule readout is not to be performed for this counter. If the sro operand is omitted, no schedule readout will be performed. This operand must not specify a test schedule greater than the highest test schedule defined by the CTRSCHED macro instruction.

cttest
: indicates which count test schedule is to be used for this particular counter. The value of the cttest operand must be from 0 to 15. A 0 value indicates that count testing is not to be done. If the cttest operand is omitted, count testing is not performed. This operand must not specify a test schedule greater than the highest test schedule defined by the CTRSCHED macro instruction.

ID
: specifies the decimal representation of the address of the area station on which this particular counter is defined. This operand is required.

SROENAB
: specifies whether or not schedule readout is to be automatically started by the 2715 at ICPL time. If SROENAB= YES is coded, this indicates that schedule readout is to be automatically started by the 2715. Coding SROENAB=NO indicates that schedule readout is not to be automatically started by the 2715; the user can initiate schedule readout

with a control request at a later time. SROENAB=YES must not be coded if the sro operand is 0 or is omitted. If the SROENAB operand is omitted, SROENAB=NO is assumed.

CTINIT
: specifies an initial count test condition that is to be started by the 2715 after an ICPL for this counter. Coding CTINIT=UNASP indicates that unassigned production testing is to be started by the 2715. Coding CTINIT=NCT indicates that no-count testing is to be started by the 2715. Coding CTINIT=NULL indicates that neither unassigned nor no-count testing is to be started by the 2715 for this counter. CTINIT=UNASP or CTINIT=NCT must not be coded if the cttest operand is 0 or is omitted. If the CTINIT operand is omitted, CTINIT=RESET is assumed. Count testing can be initiated later by a control request if it is not automatically started at ICPL time.

CTRSCHED (Counter Schedule) Macro Instruction

The CTRSCHED macro defines the count test schedules and the readout schedules to be used by all the pulse counters in the 2790 System. The count test and readout schedules may be any of 15 possible schedules in the Schedule Table.

The format of the CTRSCHED macro instruction is:

Name	Operation	Operand
[symbol]	CTRSCHED	sched,...

symbol
: The name field is optional.

sched
: specifies a count test schedule or readout schedule in minutes. This operand must be coded once for each schedule interval to be defined, but the maximum number of schedules that can be coded is 15. The value of this operand must be between 1 and 2047.

GDUTRANS (Guidance Display Unit Transaction) Macro Instruction

The GDU Transaction macro is used to build an entry in the Transaction Table. The position of the entry to be built relative to the beginning of the table is determined from the TRCODE operand. A GDUTRANS macro must be coded for every transaction code that will be used in communication with the 2798 GDUs. Each GDUTRANS macro requires two bytes of 2715 storage. The format of the GDUTRANS macro is:

Name	Operation	Operand
[symbol]	GDUTRANS	TRCODE=absexp,TRLIST=symboln

symbol
: The name field of this macro is optional.

TRCODE
: This operand is the value of a transaction code. Each transaction code is associated with a particular GDU List. The TRCODE operand is used to determine the position of an entry in the Transaction Table. The value of the TRCODE operand must be from 00 to 99. However, all values in this range need not be specified. The GDUTRANS macros must be in ascending sequence by TRCODE operands. A GDUTRANS macro found to be out of sequence or in error terminates the assembly of this macro instruction.

TRLIST
: This operand associates a GDU transaction list with the transaction code indicated in the TRCODE operand. The transaction list (TRLIST) referred to by 'symboln' is associated with the TRCODE operand.

PARAMNUM (Parameter List Number) Macro Instruction

The PARAMNUM macro is used to define an entry in the Parameter List Number Table for use with 2798 GDUs. The Parameter List Number macro is coded once for every parameter list defined by the user with the PARMLIST macro. Up to 127 PARAMNUM macros can be coded. This macro requires two bytes of 2715 storage. The format of the PARAMNUM macro is:

Name	Operation	Operand
[symbol]	PARAMNUM	PLN=absexp,PARMLST=symboln

symbol
: The name field of this macro is optional.

PLN
: This operand specifies the parameter list number that is to be associated

Name	Operation	Operand
symbol	PARMLIST	[CKLNGTH=(length-absexp, errquidance-absexp,...)] [,CKMONKY= { NO / YES }] [,CKMOD11=(length-absexp, position-absexp, errquidance-absexp,...)] [,CKRANGE=(positional1-absexp, position2-absexp, hilowchars-absexp,...)] [,LOWGUID=(absexp,...)] [,HIGUID=(absexp,...)] [,RNGETST= { ERROR / DATA }] [,CKMOD10=(length-absexp, position-absexp, errquidance-absexp,...)] [,CKOR=(position-absexp, checkchar1-hexchar,... checkcharn-hexchar)] [,ORGUID=(absexp,...)] [,CKAND=(position1-absexp, position2-absexp, checkchar1-hexchar,... checkcharn-hexchar)] [,ANDGUID=(absexp,...)] [,CKNONUM=(position1-absexp, position2-absexp, errquidance-absexp,...)] [,CKNUM=(position1-absexp, position2-absexp, errquidance-absexp,...)] [,TRANSL= { NO / YES }] [,IDENT= { NO / YES }]

with the parameter list referred to by the PARMLST operand. The value of this operand must be between 1 and 127. The PARAMNUM macros must be in ascending sequence by PLN. A PARAMNUM macro found out of sequence or in error terminates the assembly of this macro instruction. The value of the PLN operand must be 1 greater than the PLN operand of the previous PARAMNUM macro.

PARMLST
: This operand specifies the name of a parameter list defined by a PARMLIST macro.

PARMLIST (Parameter List) Macro Instruction

The PARMLIST macro is used to generate an entry in the Parameter List Table. The data in the parameter list detfines the types of checks that are to be performed on a data entry from a 2798 GDU. The entries in the Parameter List Table to be used by the 2715 are selected by a pointer from the Parameter List Number Table. Every PARMLIST macro must be referred to by a PARMLST operand in the PARAMNUM macro. The size of the Parameter List entry is variable depending on the type of tests requested. The format of the PARMLIST macro is:

symbol
The name field must be specified and must be the same name as defined by the PARMLST operand in the PARAMNUM macro.

CKLNGTH
This operand causes the 2715 to check the data entry to determine if it is the length specified.

length
This suboperand specifies the length of the data entry and its value may be from 1 to 17. Note: The first byte in the data entry is the Operational Status byte that is generated by the 2715. Therefore, the value specified by the length suboperand will always be one more than the number of characters entered by the GDU operator. For example, if the user doesn't expect any characters to be entered from the GDU keyboard he must assign a value of 1 to the length suboperand.

errguidance
This suboperand specifies the error guidance that is returned to the 2798 GDU if the data entry length is incorrect. The value of each suboperand may be from 1 to 16 and up to 16 suboperands may be coded. Each suboperand represents a light on the guidance panel that is turned on when a CKLNGTH error condition exists. (See Figure 27.1) Example: If the user coded CKLNGTH=(8,2,4,16) and the GDU operator entered any number of characters other than 7, then the error guidance lights for lines 2 and 4 on the left panel and line 16 on the right panel will be turned on.

CKMONKY
This operand indicates whether the 2715 will check to determine if the 2798 Monitor key is on. If CKMONKY=YES is coded, the 2715 checks that the Monitor key is on. If a Monitor key error is encountered when CKMONKY=YES, the error guidance, as specified by the MONERR operand in the CONFIGUR macro, is returned to the 2798 GDU.

CKMOD11
This operand causes the 2715 to perform a modulus 11 check on the field specified and tests the data in the field to ensure that it is numeric. The last position in the specified field must contain the self-check character. A detailed description of modulus 11 checking can be found in Component Description: IBM 2790 Data Communication System, GA27-3015.

length
This suboperand specifies the length of the modulus 11 check field. The value of this suboperand may be from 1 to 15. The length does not include the self-check character.

position
This suboperand specifies the starting position of the modulus 11 check field. The value of this suboperand may be between 2 and 16.

errguidance
This suboperand specifies the error guidance that is returned to the 2798 GDU if the modulus 11 check is not satisfied. The value of each suboperand may be from 1 to 16 and up to 16 suboperands may be coded. Each suboperand represents a light on the guidance panel that is turned on when a CKMOD11 error condition exists. (See Figure 27.1)

Example: CKMOD11=(6,2,2,10,14) will result in a modulus 11 check of a 7 digit field (the seventh digit is the self-check digit) starting at GDU data entry position 2. If the self-check digit does not satisfy the modulus 11 check, the error guidance lights for line 2 on the left panel and lines 10 and 14 on the right panel will be turned on.

CKRANGE
This operand causes the 2715 to check the specified field to ensure that it is neither less than the specified low test value nor higher than the specified high test value. This check also tests the data to ensure that it is numeric.

position1
This suboperand specifies the starting position of the field for which the range check is performed. The value may be from 2 to 17.

position2
This suboperand specifies the last position of the field for which the range check is performed. The value may be from 2 to 17.

hilowchars
This suboperand specifies the high and low test digits (0-9) for each character in the field. Up to 16 'hilowchars' may be coded. If the high and low test digits coded for a position of a field are the same digit, the 2715 will check that the test position is indeed that digit.

Example: CKRANGE=(8,10,91,80,63) will cause the 2715 to check for a 3 digit number starting in data entry position

8 and ending in position 10. The 3 digit number in positions 8 through 10 must be greater than or equal to 103 and less than or equal to 986.

LOWGUID
This operand specifies the error guidance that is returned to the 2798 GDU if the specified field in the CKRANGE check is lower than the low test value. The value of each suboperand may be from 1 to 16 and up to 16 suboperands may be coded. Each value represents a light on the guidance panel that is turned on when the low test condition exits. (See Figure 27.1)

Example: From the example associated with the CKRANGE operand, the user may code LOWGUID=(8,9) to inform the GDU operator if the number he entered at the 2798 GDU is less than 103. If the number is less, the error guidance lights for line 8 on the left panel and line 9 on the right panel will be turned on.

HIGUID
This operand specifies the error guidance that is returned to the 2798 GDU if the specified field in the CKRANGE check is higher than the high test value. The value of each suboperand may be from 1 to 16 and up to 16 suboperands may be coded. Each value represents a light on the guidance panel that is turned on when the high test condition exists. (See Figure 27.1)

Example: From the example associated with the CKRANGE operand, the user may code HIGHGUID=(10,12) to inform the GDU operator if the number he entered at the 2798 GDU is greater than 986. If the number is greater, the error guidance lights for lines 10 and 12 on the right panel will be turned on.

RNGETST
This operand indicates the action to be taken when the data entry fails to comply with a CKRANGE check. If RNGETST=ERROR is coded and the CKRANGE fails, the data entry is not accepted and the desired error guidance is returned to the 2798 GDU. If RNGETST=DATA is coded and the CKRANGE fails, the data entry is accepted and the desired error guidance is returned with the normal guidance for the next step. RNGETST is the only error condition for which data can be accepted.

CKMOD10
This operand causes the 2715 to perform a modulus 10 check on the field specified and tests the data in the field to ensure that it is numeric. The last position in the specified field must contain the self-check character. A detailed description of modulus 10 checking can be found in Component Description: IBM 2790 Data Communication System, GA27-3015

length
This suboperand specifies the length of the modulus 10 check field. The value of this suboperand may be from 1 to 15. The length does not include the self check character.

position
This suboperand specifies the starting position of the modulus 10 check field. The value of this suboperand may be from 2 to 16.

errguidance
This suboperand specifies the error guidance that is returned to the GDU if the modulus 10 check is not satisfied. Each value of this suboperand may be from 1 to 16 and up to 16 suboperands may be coded. Each suboperand represents a light on the guidance panel that is turned on when CKMOD10 error condition exists. (See Figure 27.1)

Example: CKMOD10=(4,1,16) will result in a modulus 10 check of a 5 digit field (the fifth digit is the self-check digit) starting at GDU data entry position 1. If the self-check digit does not satisfy the modulus 10 check, the error guidance light for line 16 on the right panel will be turned on.

CKOR
This operand causes a check by the 2715 to ensure that the character received in the position specified in the data entry is one of the check characters specified by the user. There may be one to five unique check characters associated with this test and only one must compare.

position
This suboperand specifies the position in the data entry that is checked for the character comparison. The value of this suboperand may be from 2 to 17.

checkchar1,checkcharn
Each suboperand defines a check character. From 1 to 5 of these suboperands may be coded. The value of the suboperand may be the hexadecimal

equivalent of any of the characters: A-Z, 0-9, or any of the special characters: . $ & - / , # " @ = : ? ! ; * + SPACE TAB NEWLINE LINEFEED. (See Table 9.1)

Example: CKOR=(3,D3,F5,7C,61) will check the fourth data entry position (the third character entered by the operator) to ensure that it contains one of the characters: L,5,@, or /. If the character is not one of the four specified, the user may code the following operand.

ORGUID
This operand specifies the error guidance that is returned to the 2798 GDU if the CKOR check indicates an error. The error occurs when the character in the specified data entry position does not equal any of the check characters. The value of each suboperand may be from 1 to 16 and up to 16 suboperands may be coded. Each value represents a light on the guidance panel that is turned on when a CKOR error condition exists. (See Figure 27.1)

Example: From the example associated with the CKOR operand, the user may code ORGUID=(2,3,4) to inform the GDU operator if the third character he entered at the 2798 GDU is not equal to one of the specified characters. If the character is not equal, the error guidance lights for lines 2,3, and 4 on the left panel will be turned on.

CKAND
This operand causes the 2715 to check the characters received in the consecutive positions specified to ensure that they match all of the specified check characters.

position1
This suboperand specifies the starting position of the field for which the CKAND compare is started. The value of this suboperand may be from 2 to 17.

position2
This suboperand specifies the last position of the field for which the CKAND compare occurs. The value of this suboperand may be from 2 to 17.

checkchar1,checkcharn
Each suboperand defines a check character and from 1 to 16 characters may be coded. The value of the suboperand may be the hexadecimal equivalent of any of the characters: A-Z, 0-9, or any of the special characters: . $ & - / , # @ " = : ? ! ; * + SPACE TAB NEWLINE LINEFEED. (See Table 9.1)

Example: CKAND=(7,10,C2,E3,C1,D4) will check the consecutive data entry positions 7 through 10 to ensure they contain the characters B, T, A, and M in that order. If an error occurs, the user may code the following operand. If the hexadecimal value X'00' is used as a check character, the character in the corresponding position of the data field will not be checked.

Example: CKAND=(2,5,C1,C2,00,C3) will check data entry positions 2, 3, and 5 to ensure they contain the characters A, B, and C in that order. The character in data entry position 4 will not be checked.

ANDGUID
This operand specifies the error guidance that is returned to the 2798 GDU if the CKAND check indicates an error. This error occurs when the characters received in the consecutive positions specified do not match all of the specified check characters. The value of each suboperand may be from 1 to 16 and up to 16 suboperands may be coded. Each value represents a light on the guidance panel that is turned on when a CKAND error condition exists. (See Figure 27.1)

Example: From the example associated with the CKAND operand, the user may code ANDGUID=9 to inform the GDU operator if the specified characters do not match. If the characters do not match, the error guidance light for line 9 on the right panel will be turned on.

CKNONUM
This operand causes the 2715 to check a specified field to ensure that no numeric characters are received.

position1
This suboperand specifies the starting position of the field to be checked. The value may be from 2 to 17.

position2
This suboperand specifies the last position of the field to be checked. The value may be from 2 to 17, but must be greater than or equal to the 'position1' suboperand.

errguidance
This suboperand specifies the error guidance that is returned to the GDU if a numeric character is received and a CKNONUM check is performed. The value of each suboperand may be from 1 to 16 and up to 16 suboperands may

be coded. Each value represents a light on the guidance panel that is turned on when the CKNONUM error condition exists. (See Figure 27.1)

Example: CKNONUM=(2,17,3,6) causes the 2715 to check positions 2 through 17 of the GDU data entry to ensure that all the characters are non-numeric. If any of the characters in the specified positions are numeric, the error guidance lights for lines 3 and 6 on the left panel will be turned on.

CKNUM
: This oeprand causes the 2715 to check a specified field to insure that all numeric characters are received.

position1
: This suboperand specifies the starting position of the field to be checked. The value may be from 2 to 17.

position2
: This suboperand specifies the last position of the field to be checked. The value may be from 2 to 17, but must be greater than or equal to the 'position1' suboperand.

errguidance
: This suboperand specifies the error guidance that is returned to the GDU if a non-numeric character is received and a CKNUM check is performed. The value of each suboperand may be from 1 to 16 and up to 16 suboperands may be coded. Each value represents a light on the guidance panel that is turned on when the CKNUM error condition exists. (See Figure 27.1)

Example: CKNUM=(8,8,15) causes the 2715 to check position 8 of the GDU data entry to ensure that the character in this position is numeric. If the character in the specified position is non-numeric, the error guidance light for line 15 on the right panel will be turned on.

TRANSL
: This operand indicates if the translate function will be performed on the transaction step that uses this parameter list. If TRANSL=YES is coded, the translate function will be used. If TRANSL=NO is coded, or the operand is omitted, the translate function will not be used. This operand can not be used if any other operand in the PARMLIST macro except CKMONKY is used. The TRANSI and IDENT operands are mutually exclusive.

IDENT
: This operand indicates if a store or get identifier function may be performed on the transaction step that uses this parameter list. If IDENT=YES is coded, the store or get identifier function may be used. If IDENT=NO is coded or the operand is omitted, the store or get identifer function may not be used. Other checks may be specified. The IDENT and TRANSL operands are mutually exclusive.

Note: Only three of the following seven check operands can be coded on a PARMLIST macro:

- CKMOD11
- CKRANGE
- CKMOD10
- CKOR
- CKAND
- CKNONUM
- CKNUM

The CKLNGTH and CKMONKY operands may be coded on any PARMLIST macro, regardless of how many other check operands are coded.

Only one check may be performed on a given data position in the 2798 GDU data entry. The seven check operands listed above must not overlap. A particular position in the data entry cannot be covered by more than one check. This does not apply to either the CKLNGTH or CKMONKY operands.

Left Panel	Right Panel
1	9
2	10
3	11
4	12
5	13
6	14
7	15
8	16

Figure 27.1 2798 GDU Guidance Panels

Keyboard Character	Hexadecimal Equivalent	Keyboard Character	Hexadecimal Equivalent
A	C1	2	F2
B	C2	3	F3
C	C3	4	F4
D	C4	5	F5
E	C5	6	F6
F	C6	7	F7
G	C7	8	F8
H	C8	9	F9
I	C9	TAB	05
J	D1	NEWLINE	15
K	D2	LINEFEED	25
L	D3	SPACE	40
M	D4	.	4B
N	D5	+	4E
O	D6	&	50
P	D7	!	5A
Q	D8	$	5B
R	D9	*	5C
S	E2	;	5E
T	E3	-	60
U	E4	/	61
V	E5	,	6B
W	E6	?	6F
X	E7	:	7A
Y	E8	#	7B
Z	E9	@	7C
0	F0	=	7E
1	F1	"	7F

Table 9.1. 2798 GDU Keyboard Character Conversion

DISPGUID (Display Guidance) Macro Instruction

The DISPGUID macro is used to define a display guidance message in the Display Guidance Table. A DISPGUID macro must be defined for every display guidance message the user defines. The display guidance address in the GDU step of the GDU list is used by the 2715 to address a particular display guidance message in the Display Guidance Table. The DISPGUID macro requires from 2 to 17 bytes of 2715 storage. The format of the macro is:

Name	Operation	Operand
[symbol]	DISPGUID	DISPMSG='text'[,SUPPRES= {YES / NO}]

symbol
The name field of this macro is optional.

DISPMSG
This operand defines a user specified display guidance message. The text must not exceed 16 characters.

SUPPRES
This operand indicates whether the display guidance message is returned to the 2715 after it is displayed at the GDU display guidance and the operator presses the GDU Enter Key. Coding SUPPRESS=YES or omitting the operand indicates that the defined data that was written to the GDU by the 2715 is not to be returned with the operator added data to the 2715. Only that data inserted by the GDU operator will be returned. Coding SUPPRES=NO will cause the defined data and operator inserted data to be returned, up to a maximum of 16 characters. If the operator inserted data plus the defined data exceed 16 characters, the defined data will be moved to the left and the right most characters lost.

The maximum number of DISPGUID macros that can be issued depends only on the user table size limitation.

TRANSLAT (Translate Table) Macro Instruction

The TRANSLAT macro instruction builds an entry in the Translate Table. A maximum of eight TRANSLAT macros may be coded and 3 to 16 bytes of 2715 storage are required for each. This macro is coded once for each character that is translated. The format of the TRANSLAT macro is:

Name	Operation	Operand
[symbol]	TRANSLAT	TRANSCH=hexchar,TRANTXT='text'

symbol
The name field of this macro is optional.

TRANSCH
This operand defines the character that is translated. The value for this operand is the hexadecimal equivalent of any of the characters: A-Z, 0-9, and any of the special characters: . $ & - / , # @ " = : ? ! ; * + SPACE TAB NEWLINE LINEFEED. (See Table 9.1)

TRANTXT
This operand defines the user specified translate text.

Example: An assembly line worker is required to enter the character C from a 2798 GDU each time he builds and tests a specific clutch. The programmer coded the following in the TRANSLAT macro: TRANSCH=C3,TRANTXT='4 SPEED CLUTCH'. The 2715 checks the character entered for this Translate transaction and replaces the C with the text '4 SPEED CLUTCH'. The text is now displayed at the 2798.

Note: Each character assigned to a text must be unique, that is assign a different character to each text.

TRLIST (Transaction List) Macro Instruction

The Transaction List macro is used with the Area Station List macro, the Data Entry Unit List macro, and the GDU list macro to define a transaction. When the transaction is initiated by an area station, the Transaction List macro is followed by one or more ASLIST macros. When the transaction is initiated by a data entry unit, the TRLIST macro is followed by one or more DEULIST macros. The DEULIST macros may be followed by a CTRLIST macro. When a transaction is initiated by a 2798 GDU, the Transaction List macro is followed by one or more GDULIST macros. The first transaction list must be for all of the IBM 1035 Badge Readers. This consists of a TRLIST macro instruction followed by one DEULIST macro instruction.

The Transaction Lists created by the TRLIST, ASLIST, DEULIST, GDULIST and CTRLIST macro instructions are composed of two elements: a header, and either an internal message or from one to sixteen data-entry steps. The header information is provided in the TRLIST macro instruction. The TRLIST macro must follow the last TGROUP macro.

The TRLIST macro is used to generate the transaction headers for GDU lists and can be referred to by the GDUTRANS macro.

Message routing can be specified (NULL or absexp1 coded) only when the TRLIST is part of a transaction expansion. Message routing means that explicit or explicit/implicit text is to be routed to a 1053 printer on an area station. The only ways the user can specify message routing are to code ROUTE=NULL or ROUTE=absexp1 in the TRLIST macro. Coding ROUTE=DISK, CPU, or LOG does not imply message routing, but that the transaction is to be routed to the specified ROUTE parameter. Transaction routing does <u>not</u> mean routing to an area station 1053 printer.

Each TRLIST macro requires [5 + (text length + 1)] bytes of 2715 storage. The format of the TRLIST macro is:

Name	Operation	Operand
[symbol]	TRLIST	TRID=absexp1 [,ROUTE=[{DISK | CPU}] [,LOG] [{,NULL | ,absexp2}]] [,TEXT={<u>NO</u> | YES}] [,INQDISP={<u>NO</u> | YES}] [,DEMOD10={<u>NO</u> | YES}] [,DEMOD11={<u>NO</u> | YES}] [,GDU={<u>NO</u> | YES}]

TRID=absexp1
: specifies a transaction identifier. The user assigns a value from 0 to 159 to "absexp1," and the 2715 places this value in the transaction control byte of the transaction header for priority and deferred data. The value of "absexp1" must be in ascending order with the other TRID parameters coded in the program; however, values may be omitted (a warning messages is generated at assembly time). Since the user receives the transaction header with a message, the transaction identifier allows him to determine which TRLIST macro processed the data in the 2715.

symbol
: The name field is required for this macro instruction.

ROUTE
: The ROUTE operand specifies the destination of the data records (transactions) that originate on one of the devices attached to the 2715. At least one destination must be specified, and if only one is specified the parentheses are not coded.

DISK
: specifies that the transaction should be routed to the 2715 integral disk; that is, the message is a deferred message.

CPU
: specifies that the transaction should be routed directly to the CPU; that is, the message is an inquiry or a priority message.

LOG
: specifies that the transaction is to be routed to the 2740 attached to the 2715.

NULL
: specifies that the first data entry of the transaction is the destination address of the message, that is, the hexadecimal address of an area station. The message is to be routed to the printer attached to that area station.

absexp2
: specifies the decimal representation of the address of an area station (see Table 9). The message is to be routed to the printer attached to that area station.

<u>Note</u>: The suboperands of the ROUTE parameter may be coded in any order. If one is omitted, commas need not be coded to indicate the omission.

TEXT= <u>NO</u>
YES
: specifies that a message defined in a subsequent ASLIST, DEULIST, GDULIST or CTRLIST macro is to be routed.

INQDISP
: The INQDISP operand indicates whether this transaction is an Inquiry Display transaction. Coding INQDISP=YES specifies that inquiry display will be used in this transaction. Coding INQDISP=YES requires that INQDISP=YES be coded in the CONFIGUR macro. Coding INQDIST=YES requires one extra GDULIST macro to end this transaction list. See GDULIST macro description for details.

DEMOD10
: The DEMOD10 operand indicates whether the 2715 will perform a Modulus 10 self check on all or part of a data entry from an area station or data entry unit. Coding DEMOD10=YES specifies that Modulus 10 self checking will be performed on a data entry in this transaction. Coding DEMOD10=YES requires CORE=32 to be coded in the CONFIGUR macro. This operand is mutually exclusive with the DEMOD11 operand. This operand does not apply to the 2798 GDU.

DEMOD11
: The DEMOD11 operand indicates whether the 2715 will perform a Modulus 11 self check on all or part of a data entry from an area station or data entry unit. Coding DEMOD11=YES specifies that Modulus 11 self checking will be performed on a data entry in this transaction. Coding DEMOD11=YES requires CORE=32 to be coded in the CONFIGUR macro. This operand is mutually exclusive with the DEMOD10 operand. This operand does not apply to the 2798 GDU.

GDU
: This operand allows for 100 additional transactions identifiers (TRID) to be specified by the user. The normal range of identifiers is from 0 to 159 and the additional identifiers range from 0 to 99. Coding GDU=YES resets the TRID operand checking and allows for a maximum of 100 more transaction identifiers to be specified. The checking resumes with the new identifiers which may or may not be unique identifiers. If non-unique identifiers exist, the user must also check the device address in the transaction header to determine if the transaction is for a 2798 GDU. All TRLIST macros for GDU transactions and the associated GDULIST macros should be the last macros coded before STEND. (See Appendix M)

__Example__: This example represents a series of 260 TRLIST macros with all other macros omitted:

```
TRLIST0     TRLIST     TRID=0,ROUTE=DISK
TRLIST1     TRLIST     TRID=1, ROUTE=CPU
TRLIST2     TRIIST     TRID=2,ROUTE=DISK
   .           .             .
   .           .             .
TRLST159    TRLIST     TRID=159,ROUTE=CPU
GDUTR0      TRLIST     TRID=0,ROUTE=DISK,GDU=YES
   .           .             .
   .           .             .
GDUTR99     TRLIST     TRID=99,ROUTE=CPU,GDU=YES
```

ASLIST (Area Station List) Macro Instruction

The Area Station List macro instruction is used to define one step of a transaction list for a 2791 Area Station. One to sixteen ASLIST macros may follow a TRLIST macro. If more than sixteen are used, the excess macros are flagged as errors in the assembly. Each ASLIST macro requires 5 bytes of 2715 storage. If the message operand is coded, the ASLIST macro requires additional storage of length-of-text-plus-one bytes. The format of the ASLIST macro instruction is:

Name	Operation	Operand
[symbol]	ASLIST	device-code,NORM=absexp [,LENGTH=(absexp1,absexp2)] [,DIGIT=(absexp1,absexp2,absexp3)] [,ENTRY= $\left\{\begin{matrix}\underline{1}\\ M\end{matrix}\right\}$][,MSG='text'] [,INQDISP=absexp [,MODULUS=(absexp1,absexp2,absexp3)] [,SELTRAN= $\left\{\begin{matrix}\underline{NO}\\ YES\end{matrix}\right\}$]

symbol
The name field of this macro instruction is optional.

device-code
This operand indicates the device to be activated at the 2791 Area Station. The accepted values are:

B - Badge
C - Card
M - Manual entry
O - OEM input

NORM
This operand indicates which guidance light on the area station should be switched on if no error is recognized in the previous step of the transaction (see Table 10). (The first step is considered to be the acceptance of the transaction code.) This value must be from 1 to 31.

LENGTH
absexp1 - specifies the significant length of the data entry (the number of data characters excluding blanks). This may be any value from 0 to 81; the maximum length depends on the input device -- card reader, badge reader, manual entry, OEM entry.

absexp2 - specifies which guidance light should be switched on if the number of characters received is different from the value specified by "absexp1". The value of "absexp2" must be from 1 to 31. (See Table 10.)

No length error checking takes place if the LENGTH parameter is not coded. If the LENGTH parameter is omitted, or if zero is specified, no significant length checking is done.

DIGIT
absexp1 - specifies the position or column of the value in the data entry that is to be compared with the value specified in "absexp2". The value of "absexp1" must be from 1 to 15.

Note: The actual position of the first byte of data received from the input devices depends on the device. For the local badge reader, card reader, keyboard, and OEM devices on the 2791, the first byte of data is in position 2. (Position 1 is the Monitor key.)

absexp2 - specifies a value, from 0 to 9, to be compared with a specified value in the data entry.

absexp3 - indicates which guidance light should be switched on if the specified values do not match. This value must be from 1 to 31. (See Table 10.)

If this operand is omitted, no error checking takes place.

The DIGIT operand can not be coded if DEMOD10=YES or DEMOD11=YES in the TRLIST macro.

ENTRY
This operand allows a processing step to accept multiple input data entries until the operator calls for the next step to be activated. If the ENTRY operand is omitted, or if ENTRY=1 is coded, there will be only one data entry for this step. If ENTRY=M is coded, this step may be repeated until ended by the operator. User-documented instructions to the operator must reflect the fact that the total number of bytes of data entered must not exceed the maximum transaction length of 247. This operand must not be coded on the first or only ASLIST macro following a TRLIST macro.

INQDISP
This operand specifies which guidance light on the area station is turned on when an Inquiry Display transaction is received by the 2715 and routed to the CPU as priority data. This is a user specified guidance such as "Inquiry in Process." The value of the operand must be from 2 to 31 (0 is reserved for Select Transaction and 1 is reserved for an aborted inquiry). Coding this operand requires INQDISP=YES to be coded in the TRLIST macro for this transaction.

MODULUS
This operand indicates the field in this data entry for which the 2715 performs either a Modulus 10 or Modulus 11 self check algorithm. Error guidance is also specified when the self check fails.

absexp1- Specifies the starting position of the field for the modulus check.

absexp2- Specifies the length of the field on which the modulus check is performed. The value can be from 1 to 15. This length does not include the self check character.

absexp3- Specifies which guidance light is turned on if the modulus check fails. This value must be from 1 to 31.

Note: The MODULUS operand can not be coded unless DEMOD10=YES or DEMOD11=YES is coded in the TRLIST macro. This operand is mutually exclusive with the DIGIT operand.

SELTRAN
This operand allows the Select Transaction light on the 2791 Area Station to be turned on at the completion of a transaction, instead of the first guidance light. Coding SELTRAN=YES on any ASLIST macro after the first ASLIST macro in any transaction causes the Select Transaction light to be turned on at the completion of a transaction. If the operand is omitted or if SELTRAN=NO is coded, the first guidance light is turned on at the completion of the transaction. SELTRAN=YES can not be coded on the first ASLIST macro in a transaction.

MSG='text'
Defines a user-specified message to be routed. The text must not exceed 127 characters. The destination of the message was specified in the preceding TRLIST macro instruction. This operand may only be specified for the last ASLIST macro associated with any TRLIST macro. TEXT=YES must have been coded in the TRLIST macro.

31	30	29	28
27	26	25	24
23	22	21	20
19	18	17	16
15	14	13	12
11	10	9	8
7	6	5	4
3	2	1	SELECT TRANS-ACTION
ON LINE	REPEAT CLEAR	IN PROCESS	CARD IN

Table 10. ASLIST Operand Values for Guidance Lights by Position on the Area Station

DEULIST (Data Entry Unit List) Macro Instruction

The Data Entry Unit List macro is used to define one step of a transaction list for a data entry unit or to define a transaction for the 1035 Badge Reader. For a 2796 and 2797 DEU one to thirteen DEULIST macros and for a 2795 DEU one to sixteen DEULIST macros may follow a TRLIST macro. If more than sixteen are used, the excess macros are flagged as errors in the assembly. Each DEULIST macro requires 5 bytes of 2715 storage. If the MSG operand appears, the DEULIST macro requires additional storage of length-of-text-plus-one bytes. The format of the DEULIST macro instruction is:

Name	Operation	Operand
[symbol]	DEULIST	[DIGIT=(absexp1,absexp2)] [,LENGTH=absexp1] [,MSG='text'] [,MODULUS=(absexp1,absexp2)] [,DIGIT2=(absexp1,absexp2)]

symbol
The name of the DEULIST macro is optional.

DIGIT
absexp1 - specifies the position or column of the value in the data entry that is to be compared with the value specified in "absexp2". The value of "absexp1" must be from 1 to 15. If the specified values do not match, the red error button on the data entry unit pops up, and the operator must reenter correct data.

Note: The actual position of the first byte of data received varies depending on the device. For a data entry unit (2795, 2796, 2797), the first byte of data is in position 3. Positions 1 and 2 are the Monitor key and setting of the right-hand knob. For a 1035 badge reader, the first byte of data is in position 1.

absexp2- Specifies a value, from 0 to 9, to be compared with a specified value in the data entry.

The DIGIT operand can not be coded if DEMOD10=YES or DEMOD11=YES is coded in the TRLIST macro for this transaction.

LENGTH

absexp1--specifies the significant length of the data entry (the number of data characters excluding blanks). To determine the required data entry length, use the following formulas (see Programming Notes for data entry format):

1035: Reads a badge = value from 1 to 10 or 0
2795: ID+RK+CDBD = value from 2 to 12 or 0
2796: MON+TRK+CDBD+BLK+BRK+ROCK = value from 8 to 18 or 0
2797: MON+RK+CDBD+MAN = value from 8 to 18 or 0

BLK = bottom left knob (1 byte)
BRK = bottom right knob (1 byte)
CDBD = card or badge (0 to 10 bytes)
ID = ID code (1 byte)
MAN = manual entry (6 bytes)
MON = Monitor key (1 byte)
RK = right knob (1 byte)
ROCK = digit-rocker switches (4 bytes)
TRK = top right knob (1 byte)

If zero is specified or if the LENGTH parameter is omitted, no significant length check is performed. If an invalid length is detected, the red error button pops up.

MODULUS

The MODULUS operand indicates the field in this data entry for which the 2715 performs either a Modulus 10 or Modulus 11 self check algorithm. The MODULUS operand can not be coded unless either DEMOD10=YES or DEMOD11=YES is coded in the TRLIST macro for this transaction. This operand is mutually exclusive with the DIGIT operand, but not the DIGIT2 operand.

absexp1- Specifies the starting position of the field for which the modulus check is performed. The value can be from 2 through 16 corresponding to the last data positions in the data entry.

absexp2- Specifies the length of the field for which the modulus check is performed. The value can be from 1 to 15. This length does not include the self check character.

DIGIT2

The DIGIT2 operand specifies a position in the data entry that is checked by the 2715 for a specified value. This operand can be coded when either the DIGIT operand or the MODULUS operand is coded or when neither is coded.

absexp1- Specifies the position of the value in the data entry that is compared with the value specified in "absexp2." The value of "absexp1" must be from 1 to 15. If the specified values do not match, the red error button on the data entry unit pops up and the operator must reenter the correct data.

absexp2- Specifies a value from 0 to 9 that is compared with a specified value in the data entry.

MSG

Defines a user-specified message to be routed. The text must not exceed 127 characters. The destination of the message was specified in the preceding TRLIST macro instruction. This operand may be specified only for the <u>last</u> DEULIST macro associated with any TRLIST macro. TEXT=YES must have been coded in the TRLIST macro. If a CTRLIST macro is coded, the MSG operand may be specified <u>only</u> in the CTRLIST macro.

GDULIST (Guidance Display Unit List) Macro Instruction

The GDULIST macro instruction is used to define one step of a GDU transaction list for a 2791 or 2793 Area Station with 2798 GDUs attached. One to sixteen GDULIST macros may follow a TRLIST macro. If more than sixteen are coded, the excess macros are flagged as errors in the assembly. Each GDULIST macro requires 5 bytes of 2715 storage. If the MSG operand is coded, the GDULIST macro requires additional storage equal to the length of the MSG text. If an inquiry display transaction (INQDISP=YES in the TRLIST macro) is coded, one extra GDULIST macro must be coded as the last entry of the transaction list. This macro supplies normal guidance light number and display message number only and initiates no checking or parameter list references.

The format of the GDULIST macro is:

Name	Operation	Operand
[symbol]	GDULIST	PARAMNO=absexp[,(NORGUID=absexp,...)] [,{DISPMSG=symbol / IDENT=absexp}][,MSG='text'] [,ENTRY= {1 / M}]

symbol

The name field in this operand is optional.

PARAMNO
This operand indicates the parameter list number to be used by the 2715 to get to a parameter list that defines the type of checks to be performed on the data entry for this GDU step. The value of this operand must be defined in a PLN operand of the PARAMNUM macro. The value of the PARAMNO operand must be from 1 to 127.

NORGUID
This operand indicates the normal guidance that will be sent to the operator guidance panel on the GDU when this step is entered. The value of this operand can be from 1 to 16 and up to 16 suboperands can be coded. Each suboperand represents a light on the guidance panel that will be turned on when this particular step is entered.

DISPMSG
This operand specifies the name of the DISPGUID macro that defines the message to be displayed on the 2798 Display Guidance Panel when this step in the GDU transaction is entered.

IDENT
This operand specifies an identifier in the Identifier Table to be displayed on the 2798 Display Guidance Panel when this step in the GDU transaction is entered. The value of this operand must be between 0 and 99 and must be less than the value of the IDCOUNT operand of the CONFIGUR macro (except when the IDCOUNT=0. The DISPMSG and IDENT operands are mutually exclusive. Every time the 2715 is ICPLed, the predefined text 'NOT USED' will be defined in every identifier in the identifier table. The user should use the Store Identifier function prior to using the Get Identifier function after an ICPL if he expects useful information to be in the identifier table.

Example: If the IDCOUNT operand of the CONFIGUR macro indicates there are 6 identifiers in the Identifier Table (IDCOUNT=6), and the user wants to display the fifth identifier when the step associated with a GDULIST macro is entered, then he must code IDENT=4 in this macro (IDENT=0 is the first identifier available).

Note: The user is made aware of the fact that he has not stored any text in a particular Identifier since he performed his table load by having the text 'NOT USED' defined in every Identifier in the Identifier Table at assembly time. When the user displays a particular Identifier as specified by the IDENT operand of a GDULIST macro and sees the text 'NOT USED," he should realize that he has never stored any text in the Identifier.

MSG
Defines a user-specified message to be routed. The text must not exceed 127 characters. The destination of the message was specified in the preceding TRLIST macro instruction. This operand may be specified only for the last GDULIST macro associated with any TRLIST macro. TEXT=YES must have been coded in the TRLIST macro. If a CTRLIST macro is coded, the MSG operand may be specified only in the CTRLIST macro.

ENTRY
This operand allows a processing step to accept multiple input data entries until the operator calls for the next step to be activated. If the ENTRY operand is omitted, or if ENTRY=1, there will be only one data entry for this step. If ENTRY=M, this step may be repeated until ended by the operator. User documented instructions to the operator must reflect the fact that the total number of bytes of data entered must not exceed the maximum transaction length of 247. This operand must not be coded on the first or only GDULIST macro following a TRLIST macro.

CTRLIST (Counter List) Macro Instruction

The CTRLIST macro is used to define the last step of a transaction for a data entry unit that is attached to a 2793 Area Station with pulse counters. This macro generates a five-byte data entry step for pulse count. The counter appendage step must be the last step in a transaction.

The format of the CTRLIST macro instruction is:

```
┌─────────┬──────────┬─────────────────────────┐
│Name     │Operation │Operand                  │
├─────────┼──────────┼─────────────────────────┤
│[symbol] │CTRLIST   │DEVCOD= ⎧B⎫              │
│         │          │        ⎨C⎬              │
│         │          │        ⎩M⎭              │
│         │          │CTRADR= ⎧IMP⎫,           │
│         │          │        ⎩EXP⎭            │
│         │          │CTRRD=  ⎧SINGLE⎫,        │
│         │          │        ⎩GROUP ⎭         │
│         │          │CTTEST= ⎧NULL    ⎫,      │
│         │          │        ⎪SETNCT  ⎪       │
│         │          │        ⎪SETUNAS ⎪       │
│         │          │        ⎩RESET   ⎭       │
│         │          │CTROP=  ⎧READ    ⎫       │
│         │          │        ⎪SET     ⎪       │
│         │          │        ⎪READSET ⎪       │
│         │          │        ⎪READRST ⎪       │
│         │          │        ⎪RDRESID ⎪       │
│         │          │        ⎩NULL    ⎭       │
│         │          │[,MSG='text']            │
└─────────┴──────────┴─────────────────────────┘
```

symbol
: The name field is optional.

DEVCOD
: indicates the way the data entry is entered at the DEU. If DEVCOD=B is coded, a badge will be used; if DEVCOD=C, a card will be used; and if DEVCOD=M, manual entry will be used. M may not be specified for a 2795 DEU.

CTRADR
: indicates whether implicit or explicit counter addressing is to be used. Coding CTRADR=EXP indicates that explicit counter addressing is to be used. Explicit counter addressing is entered within the last data entry. This entry is retained as data in normal transaction assembly. Addressing is specified as decimal digits with values from 1 to 63.

Coding CTRADR=IMP indicates that implicit counter addressing is to be used. Implicit counter addressing is valid only from a DEU and implies that only the first 32 counters can be used. For implicit counter addressing, the device address of the DEU initiating the request (from X'C0' to X'DF') will be converted to a counter device address (from X'1' to X'20') and used as the implied address.

CTRRD
: indicates how counters are to be read. If CTRRD=SINGLE is coded, the counters are to be interrogated individually. Coding CTRRD=GROUP indicates that counters are to be interrogated on a group basis. Group reads are done on a from/to basis with a 16-counter maximum.

CTTEST
: specifies the count test options. Coding CTTEST=NULL indicates that there is no change in the present count test condition. Coding CTTEST=SETNCT indicates that no-count test will be enabled and the unassigned production test will be disabled. Coding CTTEST=SETUNAS indicates that the no-count test will be disabled and the unassigned production test will be enabled. Coding CTTEST=RESET disables all testing conditions.

CTROP
: indicates the type of counter request to be performed. Coding CTROP=READ indicates that the counters are not to be reset after a single or group read. Coding CTROP=SET indicates that the counters are to be set to the value specified by the user at the DEU. Coding CTROP=READSET indicates that the counters are set to the value specified by the user at the DEU after a single or group read. SET and READSET are valid only for 2796 and 2797 DEUs and may not be routed to the ASLOG printer. Coding CTROP=READRST indicates that the counters are to be reset to zero after a single or group read. Coding CTROP=RDRESID indicates a read residual function, after which the counters are not reset. Coding CTROP=NULL indicates that no read or set counter functions will be performed in this transaction.

MSG
: defines a user-specified message to be routed. The text must not exceed 127 characters. The destination of the message was specified in the preceding TRLIST macro instruction. This TRLIST macro must also have specified TEXT=YES.

STEND (Statement End) Macro Instruction

The Statement End macro instruction is used with or without a name and must have no operands. It is used to indicate the end of all user macros. This must be the last card processed before the assembler END card. The STEND macro instruction compares the total number of bytes generated for the 2715 tables with the maximum allowable size for the user's particular 2715 (see CONFIGUR). If the size of the tables exceeds the maximum, an MNOTE is issued indicating the assembly is invalid.

Name	Operation	Operand
[symbol]	STEND	

symbol
: The name field of this macro is optional.

Note: A warning MNOTE is generated by this macro.

PROGRAMMING NOTES

The following general operational characteristics should be remembered when communicating with a 2715:

- When priority data has been read to exhaustion (EOT received), the user should write a control message to the 2715 requesting deferred data and then read that data until an EOT is received.
- When a data message has been accepted by the 2715 but cannot be routed to its ultimate destination, the 2715 sends the message back to the System/360 with the transaction control byte unchanged, and an error code in the zone field of the second byte of the time field of the transaction header. The error codes are:

2740 not attached	1110
2740 not operational	1101
Incomplete transaction	1100
1053 not attached	1011
1053 not operational	1010
2740 overload	1001
MSG routine overload	1000
Invalid request from CPU	0111
Counter not attached (Pulse Count feature)	0101
Device not operational (Pulse Count feature)	0011

- The devices attached to a 2790 system may vary in their ability to transmit blanks. This may affect the length of data entered, that is, data from local card reader, badge reader, etc.

- A 2715 data entry consists of the following:

2795 Data Entry Unit

LEFT KNOB	ID CODE	RIGHT KNOB	10 DATA BYTES
	12 BYTES		

2796 Data Entry Unit

TOP LEFT KNOB	MON-ITOR KEY *	TOP RIGHT KNOB	10 DATA BYTES	BOT-TOM LEFT KNOB	BOTTOM RIGHT KNOB	4 DIGIT ROCKER SWITCH-ES
	18 BYTES					

* Has a value of 1, 2, or 3.

2797 Data Entry Unit

LEFT KNOB	MON-ITOR KEY **	RIGHT KNOB	10 DATA BYTES	6-DIGIT MANUAL ENTRY BUFFER
	18-BYTES			

** Has a value of 4, 5, or 6.

2798 Guidance Display Unit

TRANS-ACTION CODE BYTE	OPERA-TIONAL STATUS BYTE	MAXIMUM OF 16 DATA BYTES
	17 bytes	

2791 Area Station

TRANSACTION CODE	MONITOR KEY*	1 TO 80 DATA BYTES**
	1 TO 81 BYTES	

*Not included with data entries from 1035 Badge Readers
X'F0' = Key off
X'F1' = Key on
**Card reader-80 bytes
Badge reader-10 bytes
Manual entry-6 bytes
OEM entry-10 bytes

The Monitor key on an area station or a data entry unit allows the operator to add an approval to a given transaction. Approval is accomplished through the transmission of a unique character that is activated by placing a key in a two-position lock switch for the 2791 Area Station and a three position lock switch for a 2796 or 2797 Data Entry Unit.

Note: The 2715 removes the first character, which is the transaction code (from a transaction key on an area station or the value of the left-hand knob on a 2795 or 2797 or of the top left-hand knob on a 2796 Data Entry Unit).

- The user must provide input/output areas or buffers of at least 640 contiguous bytes to allow for the maximum message length that can be received from the 2715.

- A separate assembly of the following macros is required for table generation:

  CONFIGUR
  AS
  TGROUP
  ASCTR (optional)
  CTRGROUP (optional)
  CTRSCHED (optional)
  TRLIST
  ASLIST (DEULIST)
  CTRLIST (optional)
  STEND

- The first assembly statement must be the CONFIGUR macro (there must not be a TITLE, CSECT, or START card). The last assembly statement must be the END card.

Notes:

1. When transaction expansion is specified, all TRLISTs referred to by this group must be such that the device selection and normal guidance in the first data entry of each of these transaction lists are identical.

2. Storage expansion (32K) is a prerequisite to transaction expansion.

3. Transaction expansion is a prerequisite to message routing.

4. A transaction expansion entry must not refer to another transaction expansion entry.

5. The first transaction must be for all of the IBM 1035 Badge Readers.

6. Chaining data entries is not allowed for the IBM 1035 Badge Reader.

7. The value coded in the LENGTH parameter must be equal to the number of data characters (nonblank) plus 1.

8. MSG operand may only be specified for the last ASLIST, DEULIST or CTRLIST

macro associated with any TRLIST macro.

9. The last entry of a transaction cannot be a multiple entry.

10. The maximum transaction length on a multiple entry is 247 bytes.

11. All DEUs attached to an area station <u>must</u> use a common transaction group (TGROUP). If 2795, 2796, and 2797 DEUs are attached to the same area station, three TGROUP macro instructions must be coded, but only one DEGROUP operand is coded in the AS macro for this area station. See Figure 27 for examples and the discussion of the TGROUP macro instruction for details.

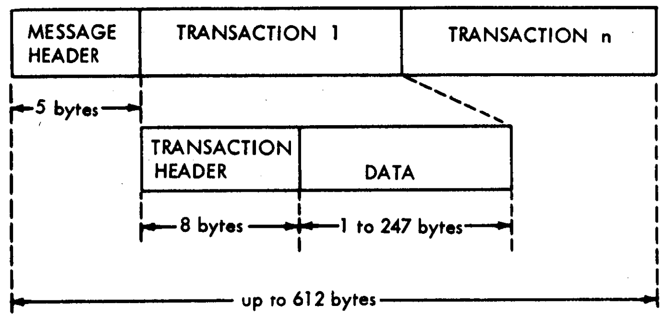
MESSAGE FORMAT

The user communicates with the 2715 using BTAM READ and WRITE macro instructions and BSC line control procedures. When reading from the 2715, the length of the message is text length plus 3 (DLE STX is received at the beginning of the message and ETX is received at the end). The maximum length for text received is 640 characters.

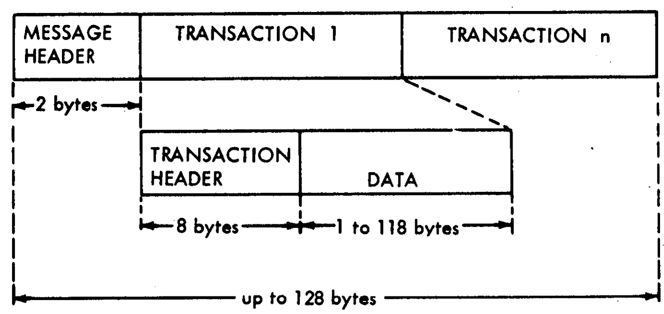
When writing to the 2715, the number of bytes coded in the length operand of the WRITE macro instruction is text length plus 2 (the user must insert DLE STX in front of the text). The total number of bytes written is text length plus 4 (BTAM inserts DLE ETX at the end of the text). The maximum length for text written is 128 characters.

Each message transmitted or received is composed of one or more transactions, preceded by a message header. Each of the transactions is composed of a transaction header and data. When transmitting to the 2715, these headers must be provided by the user in correct format.

Message formats are shown in Figure 28.



2715 to System/360



System/360 to 2715

Figure 28. Message Formats

<u>Message Header -- System/360 to 2715</u>

The message header is two bytes and has the following format:

Byte 0 Message length

Byte 1 Message control byte

<u>Message Length</u>: The message length is a one-byte count, in hexadecimal, of the number of characters in the message, including headers and data. The BSC framing control characters are not included in this count. For transmission from the System/360 to the 2715, the message length should not exceed 128 bytes.

<u>Message Control Byte</u>: The message control byte is used to indicate one of three possible destinations for output data, as shown below.

MESSAGE DESTINATION	CONTROL BYTE
1053 Printer or Pulse Count	X'01'
2715 Control	X'02'
2740 Terminal	X'04'

Message Header -- 2715 to System/360

The message header is five bytes and has the following format:

Bytes 0 - 2	Work day number
Bytes 3 - 4	Restart number

Work Day Number: The work day number is a three-byte EBCDIC field used as a date field. The date may be omitted,in which case the field is undefined.

Restart number: The restart number is a two-byte field that defines the type of data and associated restart information. The format of this field is:

Byte 3:	bits 0-4	Low-order bits of cylinder address.
	bit 5	Track.
	bits 6-7	Sector.
Byte 4:	bit 0	Reserved.
	bit 1	If on, indicates deferred data.
	bit 2	If on, indicates priority data.
	bits 3-4	Reserved.
	bits 5-7	High-order bits of cylinder address.

Transaction Header

The transaction header is eight bytes and has the following format:

Byte 0	Transaction length
Byte 1	Transaction control byte
Byte 2	Area station address
Byte 3	Device address (counter address)
Bytes 4 - 7	Time stamp

Transaction Length: For transmission from the 2715 to the System/360, the transaction length is a hexadecimal count of the number of bytes in a transaction, including the header. The count may not exceed 255; therefore, the maximum number of bytes of data is 247.

For transmission from the System/360 to the 2715, the transaction length is a user-provided hexadecimal count of the number of bytes in a transaction, including the transaction header. The count must not exceed 126; therefore, the maximum number of bytes of data is 118. The 2715 checks the summation of all transaction lengths against the message length. If they do not agree, the 2715 transmits an EOT, aborting the transmission.

Transaction Control Byte: The transaction control byte is a binary code that specifies the type of transaction. Values for the transaction control byte are shown in Table 11. If the value in a control transaction is not recognized by the 2715, a message is returned to the System/360.

Note: When a data message has been accepted by the 2715 but cannot be routed to its ultimate destination, the 2715 sends it back to the System/360 with an error code inserted in the zone field of the second byte of the time stamp. See the Time Stamp description for definition of the error codes. The transaction control byte still contains the "System/360 to 2715" indication.

Area Station Address: For transmission from the 2715 to the System/360 the Area Station Address field usually contains the area station address. The field contains a hexadecimal value (see Table 9).

- For priority data and deferred data, the field contains the area station address.
- For responses to control transactions, the field is undefined.

For transmission from the System/360 to the 2715, the field normally contains the area station address. The user must specify the address in hexadecimal (see Table 9). When the field contains an invalid area station address, the transaction is returned intact with an error code (see Programming Notes). The area station address field should be zero for control or 2740 operations.

Device Address: For transmission from the 2715 to the System/360 the Device Address field identifies the sending data entry unit, 1053 Printer, 1035 Badge Reader, OEM device, or 2791 resident card, badge, or manual entry, or the actual counter address. This field is zero if it is control information. Addresses in this field are represented in hexadecimal form.

For transmission from the System/360 to the 2715, the field usually contains the address of the printer on the area station. The field is zero for the 2740 or control transactions.

Time Stamp: The time stamp is a four-byte field that contains the value of the clock when the data was received. It is carried in conventional form, in hours and minutes, as EBCDIC characters. The field may be omitted on output to the 2715. If the field is omitted, four zero EBCDIC characters (X'F0') must be inserted.

An error condition will be encoded into the zone bits of the second byte to preserve the original time stamp. Note that the zone bits of the first time byte may also be changed.

The following error codes are assigned:

X'E' 2740 not attached -- The 2740 is not attached to the system, and the 2740 was specified in a user table entry. The transaction-list number in the header identified the incorrect user-table entry.

X'D' 2740 intervention required -- The 2740 requires intervention because it has power off, is out of paper, or is in improper mode.

X'C' Incomplete transaction -- This transaction is incomplete due to one of several causes:

- Operator aborted the transaction.
- Byte count was exceeded on a repeat transaction.
- Stop loop was executed and transaction was not completed in the time allowed.
- Incomplete communication with a counter for any request.

X'B' 1053 not attached -- The transaction was addressed to an area station that did not have a 1053 attached. This can be due to CPU program problems if the CPU originated the transaction, due to user table problems in the case of message routing with implicit addressing, or due to operator errors in message routing with explicit addressing.

X'A' 1053 not operational -- The addressed station has a 1053 attached, but for some reason it is not operational.

X'9' 2740 overload -- The 2740 was specified in so many transactions that a significant part of 2715 buffering was queued for the 2740 and system operation was affected. In this case, the 2715 will flag transactions with this error code, bypass the 2740, and send them to the processor as priority data. Transactions already on the 2740 queue are not affected and print out at the 2740. When the 2740 queue clears, the system will revert to its normal operation.

X'8' Message-routing overload -- The output queue contained so much of the 2715 buffering that system operation was affected. The 2715 will flag transactions with this code and route them to the processor as priority data. Transactions already on the output queue are handled normally. When the output queue clears, the system will return to normal operation.

X'7' Invalid request from CPU -- The original transaction from the CPU is returned to the CPU due to one of the following causes:

- An improper command.
- Wrong transaction length.
- An invalid value specified from a DEU to set a counter (Pulse Count feature).

X'5' Counter not attached -- The counter specified is not present on the area station (Pulse Count feature).

X'3' Device not operational -- The specified counter is present, but is not operational due to a busy condition in the counter adapter (Pulse Count feature).

Note: In the X'8' and X'9' cases, the user program still has access to the 2740 or 1053 output within normal output limitations. Thus the user may reroute this traffic under control of his program as he wishes.

It is the user's responsibility to restore the zone bits in the first and second bytes of the time-stamp field whenever he detects an error if he wishes to restore the time field to true EBCDIC representation (for example, if he were to reroute the transaction).

Data with 2798 Transactions

The first byte of data of every step in a transaction from a 2798 indicates whether the monitor key was on or off at the 2798 from which the transaction was entered. This monitor key byte is either X'FA' indicating monitor key off, or X'FB' indicating monitor key on. Following the monitor key byte can be a data field containing from 0 to 16 data characters received from the 2798 GDU. When the user is analyzing a 2798 transaction, he can separate each step by comparing for a X'FA' or X'FB' (or both) in the transaction depending upon whether or not he expects the monitor key to be on or off.

Data with Counter Control

The first byte of data is the counter control byte indicating the type of pulse counter operation this transaction results from. The counter control byte can be the response to any of the read functions (Read, Read Residual, Read and Reset, Read and Set, Read Group), or one of the following operations:

Counter Control Operation	Counter Control Byte (in Hex)
Scheduled Readout	F4
No-Count Test Failure	F2
Unassigned Production Test Failure	F1
Overflow Interrupt	F6
Power Interrupt	F7
Invalid Transaction From AS	F0

The second byte of data always contains a blank character (X'40'). The third and fourth bytes contain the address of the counter that the operation resulted from. In the case of the Read Group operation, the third and fourth bytes contain the address of the first counter in the group. The fifth byte of data is another blank character. The next five bytes contain the counter value. Except for the following operations, there is no more data in the data area.

- Read Group.
- Read Group Residual.
- No-Count Test Failure.
- Unassigned Production Test Failure.
- Overflow Interrupt.
- Power Interrupt.

For the Read Group and Read Group Residual operations, all the remaining counter values are contained in the data area, and each is separated by a blank character (X'40'). For other operations, the following EBCDIC messages are in the data area, preceded by a blank character:

OPERATION	Message
No-Count Test Failure	NCTF
Unassigned Production Test Failure	UPTF
Overflow Interrupt	OVFL
Power Interrupt	POWR

Message Type and Function	Control Byte	Data Entry Bytes $(d_1, d_2, \ldots d_n)$
System/360 to 2715		
- Data transactions		
- 1053 printer data	X'FA'	d_1-d_n (max=118 bytes)
- 2740 terminal data	X'FB'	d_1-d_n (max=118 bytes)
- Control transactions		
- Bypass area station; causes the specified area station to go offline.	X'C1'	d_1=area station address in hexadecimal
- Restore area station; causes the specified area station to go online.	X'C2'	d_1=area station address in hex
- Bypass segment; causes the specified segment of the transmission line to be bypassed.	X'C3'	d_1=segment to be bypassed in EBCDIC
- Restore segment; causes the specified segment of the transmission line to be restored to operation.	X'C4'	d_1=segment to be restored in EBCDIC
- Read deferred data; causes deferred data to be sent from the 2715 disk to the System/360.	X'C5'	none
- Stop 2790 input.	X'C6'	none
- Start 2790 input.	X'C7'	none
- Alarm messages:		
• Text; sends user error message to the area station 1053.	X'CD'	d_1-d_n=user error message in EBCDIC
• Alarm; causes alarm bell to ring at the area station 1053.	X'CE'	none
• Alarm and text; sends alarm and user error message to area station.	X'CF'	d_1-d_n=user error message in EBCDIC
- User table load start.	X'D1'	none
- User table load data; defines the data that follows as 2715 tables.	X'D2'	d_1-d_{80}=one object card from the user's assembly of 2715 macro instructions
- User table load end.	X'D3'	none
- CPU restart; recovers deferred data that was received subsequent to the specified restart number (used with the checkpoint/restart capability)	X'D4'	d_1-d_2=restart number that was checkpointed (in hexadecimal)
- 2715 restart; attempts to recover data that has been buffered at the 2715 and not yet transmitted to the System/360 after a 2790 or 2715 irrecoverable error or stop	X'D5'	d_1-d_2=deferred restart number log (in hexadecimal) d_3-d_4=priority restart number log (in hexadecimal)
- Sort area station errors; causes the 2715 to scan the error logout file and extract error statistics for the address specified in the fourth byte of the transaction header.	X'D6'	none
- Read partial error log; causes error data in the 2715 error logout file to be transmitted to the System/360.	X'D7'	none
- Reinitialize 2715 disk (will not be initiated unless all 2790 input is stopped and all deferred data is transmitted to the System/360).	X'D8'	none

Table 11. Transmission Control Byte Usage (Part 1 of 2)

Message Type and Function	Control Byte	Data Entry Bytes ($d_1, d_2 \ldots d_n$)
- Set day stamp.	X'E2'	none
- Monitor day number; causes the previously set day stamp to be monitored.	X'E3'	none
- Monitor time; causes Real-Time clock to be monitored.	X'E4'	none
- Reset deferred data mode; causes the 2715 to stop queuing deferred data from the disk to be transmitted to the System/360 (the 2715 will continue to transmit the deferred data already queued).	X'E6'	none
- All Pulse Count transactions:	X'FC'	d_1=counter control byte in EBCDIC
• Read Group functions; reads the contents of one or more counters.		d_2-d_3=EBCDIC value of last counter in the group
• Set functions and Read and Set functions; sets the counter to a predetermined count.		d_2-d_6=EBCDIC value to which the counter is to be set.
2715 to System/360		
- Data Transactions	X'00' X'7F'[1]	
- Control Transactions		
- Positive response to CPU request	X'CA'	
- Negative response to CPU request	X'CB'	
- Invalid response to CPU request	X'CC'	
- Positive response to 2715 request	X'DA'	
- Negative response to 2715 request	X'DB'	
- Response to invalid 2715 request	X'DC'	
- User defined	X'F0' X'F9'	
- Automatically initiated response	X'FF'	
- Unsolicited 2715 response	X'FD'	
- Special pulse counter transactions		
- Positive response to CPU request	X'CA'	d_1=counter control request d_2=control definition
- Invalid response to CPU request	X'CC'	same as X'CA'
- Positive response to operator initiated pulse counter request	X'DA	d_1=counter control request d_2=control definition
- Pulse count transactions for CPU	X'ED'	
- CE-initiated response for pulse counters	X'EF'[2]	

[1]This is the value of the transaction identifier.

[2]A CE can run diagnostics on a counter or counters for a specific area station while the system is still active with normal customer transactions. If the user wants to save the counter values, the CE can issue a diagnostic code to route all the counter values to the System/360. After the CE has returned the counters to the system, they may be restored to the original values by the user program.

Table 11. Transmission Control Byte Usage (Part 2 of 2)

CONTROL TRANSACTIONS

Control transactions are formatted the same as other types of transactions, using the transaction control byte of the transaction header to indicate the action to be performed. The data field of the message is used to identify the specific object of the action, for example, the identification number of the area station to be restored (the data field may or may not be present, depending on the nature of the transaction control type).

The control transaction types, as presently defined, are:

- System/360 to 2715 (sent by the user program):

  Bypass area station.
  Restore area station.
  Bypass segment.
  Restore segment.
  Stop 2790 input.
  Start 2790 input.
  User table load start.
  User table load data.
  User table load end.
  CPU restart.
  2715 restart.
  Sort area station errors.
  Read partial error log.
  Reinitialize disk.
  Set day stamp.
  Monitor day number.
  Monitor time.
  Reset deferred data mode.
  Read deferred data.
  All pulse count transactions.
  Alarm.
  Text.
  Alarm and Text.

- 2715 to System/360 (sent to user program):

  Positive response to CPU request.
  Negative response to CPU request.
  Response to invalid CPU request.
  Positive response to 2715 request.
  Negative response to 2715 request.
  Response to invalid 2715 request.
  User defined.
  Automatically initiated response.
  CE-initiated response.
  Unsolicited 2715 response.
  Pulse count responses to 2715 operator-initiated requests.
  Pulse count transactions destined for CPU
  CE-initiated response for pulse counters.

In addition to the above transactions, there are two types of messages that are transparent to the user (i.e., non-user data).

1. Error records are recorded by BTAM on a disk file; and
2. Diagnostic information (automatic or resulting from Customer Engineer intervention at the 2715 local) is printed by BTAM on the System/360 Console or the 2740, if available. Diagnostic information from the 2715 remote goes to the 2740 Data Communications Terminal.

Pulse Count Transactions

All pulse count transactions initiated from a System/360 have a control byte of X'FC' in byte 1 of the 8-byte transaction header. Byte 2 contains the area station address and byte 3 the counter address (in hexadecimal). Particular kinds of pulse counter operations are specified in the transaction text or data. The first byte of the transaction text is the counter control byte. This byte specifies the counter operation requested. Only one data byte (the counter control byte) is required for all counter operations except the Set functions, the Read and Set functions, and the Read Group functions.

For the Read Group and Read Group Residual operations, two additional data bytes must follow the counter control byte. These two bytes are the EBCDIC value of the last counter in the group. The upper limit of the last counter is 63, since there can be only 63 counters on a single area station.

For the Set functions and the Read and Set functions, five additional data bytes must follow the counter control byte. These five bytes contain the EBCDIC value to which the counter is to be set. The value must be between 0 and 29,999 in EBCDIC. These five additional data bytes are required for the following operations:

- Set Counter
- Set Counter and Set No-Count Testing and Reset Unassigned Production Testing
- Set Counter and Reset No-Count Testing and Set Unassigned Production Testing
- Set Counter and Reset all count testing functions
- Read and Set
- Read and Set and Set No-Count Testing and Reset Unassigned Production Testing
- Read and Set and Reset No-Count Testing and Set Unassigned Production Testing
- Read and Set and Reset all count testing functions

The counter control operations and the hexadecimal representation of the counter control bytes are shown in the following table:

Counter Control Operation	Counter Control Byte (in hex)
Set no-count testing (NCT), reset unassigned production testing (UNASP)	01
Reset NCT, set UNASP	02
Reset all count testing functions	03
Set counter	20
Set counter, set NCT, reset UNASP	21
Set counter, reset NCT, set UNASP	22
Set counter, reset all count testing functions	23
Read counter	80
Read counter, set NCT, reset UNASP	81
Read counter, reset NCT, set UNASP	82
Read counter, reset all count testing functions	83
Read residual	A0
Read residual, set NCT, reset UNASP	A1
Read residual, reset NCT, set UNASP	A2
Read residual, reset all count testing functions	A3
Read and reset counter	C0
Read and reset counter, set NCT, reset UNASP	C1
Read and reset counter, reset NCT, set UNASP	C2
Read and reset counter, reset all count testing functions	C3
Read and set counter	E0
Read and set counter, set NCT, reset UNASP	E1
Read and set counter, reset NCT, set UNASP	E2
Read and set counter, reset all count testing functions	E3
Read group	88
Read group residual	C8
Disable schedule readout (single counter)	B0
Enable schedule readout (single counter)	B1
Disable schedule readout and count testing (all counters on an AS)	B2
Enable schedule readout and count testing (all counters on an AS)	B3
Disable all schedule readouts (on 2790 System)	B4
Enable all schedule readouts (on 2790 System)	B5
Disable all count testing functions (on 2790 System)	B6
Enable all count testing functions (on 2790 System)	B7
Disable all schedule readouts and count test functions (on 2790 System)	BB
Enable all schedule readouts and count test functions by continuing from stop point	BC
Enable all schedule readouts and count test functions by re-initializing all schedules	BD

Table 12. Counter Control Operations

Overflow Interrupt

An overflow interrupt message is transmitted to the user-defined routing indication (specified in the ASCTR macro) whenever any counter reaches a value of 30,000.

Power Interrupt

The reporting of initial power-up or power failure at an area station results in a power interrupt message being transmitted to the user-defined routing indication (specified in the ASCTR macro). Until the power interrupt is reported from the area station, all counter transactions will be incomplete transactions.

EXTERNAL ALARM CONTACT FEATURE

The Area Station External Alarm Contact feature is provided as a method of alerting the operator at the area station level that an alarm condition exists in his area. This feature on a 2791-1 or 2793-1 Area Station allows the attachment of an external device at the area station 1053 printer, which can make use of a contact closure to operate some kind of external alarm whenever the EBCDIC character for BELL (X'2F') is received at the area station 1053 printer.

Three types of alarm messages can originate from either the System/360, the 2740 attached to the 2715, or an area station or data entry unit.

1. Alarm

2. Text

3. Alarm and text

The alarm message causes the 2791/2793 alarm hardware to be activated. The text message consists of data that is printed on the 1053 printer. The alarm and text message consists of data that causes the 2791/2793 alarm hardware to be activated and that causes the data to be sent to the 1053 printer. If the 1053 is not available, alarm or alarm and text messages are routed to the CPU. The 2791/2793 alarm hardware is activated for the alarm or alarm and text messages whether or not the 1053 printer is available. Text messages initiated at the System/360 or 2740 must be supplied by the user with the transaction request. Area station and data entry unit requests may have text supplied as explicit or implicit text.

The alarm messages initiated from the System/360 are handled as normal System/360 to area station 1053 printer output messages. The transaction control byte defines the type of alarm message, as follows:

Control Byte Value	Alarm Message Type
X'CD'	Text
X'CE'	Alarm
X'CF'	Alarm and text

The data can be any normal user data. For alarm or alarm and text messages, the 2715 generates the alarm character to send to the 1053 (the user does not have to do this).

The alarm message initiated from the 2740 is handled as a special control request. This request must be coded as follows:

BID	D1	D2	D3	Text	EOT

where:

BID is the 2740 BID key.

D1 is the type of request and can have the following values:

Y for alarm message.
Z for text message.
X for alarm and text message.

D2 and D3 represent the area station address to which the message is to be sent (decimal 00 to 99).

Text is any user text up to 127 characters.

EOT is the 2740 EOT key.

Messages originating at an area station or data entry unit are initiated by an input transaction in conjunction with the 2715 user tables. The implicit or explicit area station address, if other than the area station address of the originating station, indicates that two messages will be created by the 2715. One message will be the alarm message that will be sent to the area station than initiated the transaction. The second message will contain the data to be routed to some other area station 1053 printer. This second message will normally not be an alarm message. If, however, the user desires to send alarm and text to another area station, the first two characters of the text must be the alarm, text, or alarm and text characters. If the implicit or explicit area station address is the address of the transaction initiator, then only an alarm message will be generated by the 2715. Message routing and implicit text will be specified in the transaction list header for messages originating at an area station or data entry unit. The first two bytes of implicit text must define the type of alarm message desired. For alarm messages, these bytes will be deleted from text sent to the 1053. The following table shows the format of the first two bytes of the implicit text for alarm messages. Implicit text is specified in the 2715 user tables in the MSG operand of the last ASLIST or DEULIST macro or the CTRLIST macro in a defined transaction. The user must multipunch a 0-7-8-9 sequence for the EBCDIC BELL character specified in the table. The SPACE character indicates that nothing is punched in this column of the card.

Message Type	First Byte	Second Byte
Alarm	BELL	SPACE
Text	SPACE	BELL
Alarm and Text	BELL	BELL

2740 TRANSACTIONS

The 2740 Communications Terminal is a standard feature on the 2715 remote and an optional feature on the 2715 local. It is used with the 2715 as both an input (inquiry, control) and output (response, error logging) device.

A message originated by the System/360 and destined for the 2740 terminal has a maximum length of 128 bytes (two-byte message header, eight-byte transaction header, 118 bytes of text). These messages have the following format:

- Bytes 1 and 2: Message header (these bytes are not printed on the 2740).
- Bytes 3 through 10: Transaction header (Bytes 4, 5, and 6 are printed in hex. Following these six printed characters [two for each byte] is a space. Bytes 7 through 10 are printed as they appear in main storage. There will be a total of 11 characters printed, including the space.)
- Data: The maximum length of the actual data text is 118 bytes if the margins of the 2740 are set to maximum printing space.

A message entered from the 2740 may be formatted by the 2715 as a normal 2740-initiated request and routed directly to the System/360 user or will be treated as a control request. The 2715 will format a standard eight-byte transaction header, inserting the first character entered from the keyboard in the control byte of the header. If this byte is numeric, a four-byte time stamp will be added, subsequent characters from the 2740 will be inserted as text, and the message will be routed to the System/360.

If the first character entered from the 2740 is not numeric, the 2715 will not add a time stamp and will treat the message as a control request.

SYSTEM CONFIGURATION

An IBM 2770 can communicate with a System/360 over a nonswitched line (point-to-point or multipoint) or a switched line. The 2772 Control Unit must be equipped with the Multipoint Data Link Control feature for use on a multipoint line. A control unit for use on a switched line can be equipped with an Automatic Answering feature, if desired.

TRANSMISSION CODES

The IBM 2770 communicates with the System/360 using either of two transmission codes, EBCDIC or USASCII, as selected when the 2770 is ordered. If the 2770 is equipped with the EBCDIC Transparency feature, text data can contain any of the 256 EBCDIC bit patterns. That is, when text data is sent in transparent mode, the EBCDIC bit patterns representing data link control and terminal control characters are treated simply as data, and do not cause the control functions usually effected by these bit patterns to occur. This feature allows transmission of various kinds of raw data, such as packed decimal numbers, floating-point numbers, and machine-language programs. When transmission is in nontransparent mode, however, the data link and format control characters are recognized as such, and thus cannot appear as normal text.

TERMINAL POLLING AND SELECTION

In order to activate a terminal so that data transmission can occur, the central computer transmits on the communications line a specific character sequence that identifies the input or output component (and in the case of multipoint lines, the terminal as well) from which data is to be received, or to which data is to be sent. This procedure is called polling when an input device is involved, and selection when an output device is involved; the character sequences are called polling sequences and selection sequences. Specific polling and selection sequences are assigned to 2772 Control Unit input and output adapters, rather than to specific device types, as is the case with some other terminal types. The specific adapter-to-device correspondence is established by the customer engineer when the 2770 system is installed. You must be aware of the correspondence in order to select correctly the character sequences you need. These sequences are given below.

Point-to-Point

In point-to-point communication, you may perform component selection by one of two methods. The output device may be selected by the Job Select Switch on the 2770 operator control panel, or it may be selected by the transmission of a device control character. If more than one device is assigned by the Job Select Switch, the device control character is mandatory. DC1, DC2, and DC3 are the device control characters for output devices attached to output adapters 1, 2, and 3.

Device control characters for point-to-point lines may be sent as separate message blocks, or accompanied by text data, in the format STX DCx [text] ETB, or STX DCx [text] ETX. You code this message block in the output area referenced by a WRITE macro instruction. A device control character can be sent only as a nontransparent block, and it must be the first block of a message, i.e., following EOT or the first block of conversational reply.

The polling function is not used for a 2770 on a point-to-point line, since message transmission from terminal to computer is initiated only by the terminal.

Multipoint

Message transmission between computer and 2770 via a multipoint line is intiated only by the computer, using a Read Initial or Write Initial operation. You code polling and selection sequences in terminal lists (called polling lists for polling sequences, and addressing lists for selection sequences). (The terms selection and addressing are used here synonymously.) The DFTRMLST macro instruction is used to create the terminal lists. The READ or WRITE macro instruction that initiates message transmission sends the polling or selection sequences contained in the list.

Polling and selection sequences consist of four characters. The first is the terminal address, which may be any alphabetic character; it identifies an individual terminal and is set by the customer engineer when the terminal is installed. In a polling sequence, this character must appear in uppercase, e.g., A. When in a selection sequence, it must be lowercase, e.g., a.

The second character is always identical to the first. The third character in the sequence is a component polling or selection character. The characters DC1, DC2, and DC3 select the output devices attached to output adapter 1, 2, and 3, respectively. The characters 5, 6, and 7 poll the input devices attached to input adapters 1 (keyboard), 2, and 3; 0 causes a general poll, resulting in receipt of data from any ready input device.

The fourth character in the sequence is always ENQ (inquiry), which elicits a response from the terminal control unit that indicates whether the polled or selected component is ready.

TEMPORARY TRANSMISSION DELAYS

In communication between the IBM 2770 and the central computer, message transmission may need to be delayed because of conditions at the 2770. The 2770 signals the central computer that delay is necessary by sending one of several data link control sequences, the specific one depending on the reason for the delay. These sequences, and the automatic BTAM response or appropriate user program response to each, are as follows.

2770 Unable to Send (STX ENQ)

When during transmission of text from the 2770 to the central computer the 2770 becomes temporarily unable to transmit, it sends an STX ENQ sequence in lieu of text. Upon receiving this sequence, BTAM automatically transmits a NAK character. Transmission of STX ENQ and NAK alternates until (1) the 2770 once again is able to transmit, in which case it resumes transmitting text (or EOT, if there is no more text to transmit), or until (2) the BTAM retry count of seven is reached. In the latter event, BTAM turns on bit 7 of DECFLAGS and posts a completion code of X'41'. When this happens, the user program should ordinarily transmit an EOT and reestablish contact later.[1]

2770 Unable to Receive Text (WACK)

At the time the 2770 receives an ENQ or selection characters from the central computer, or after it has already received one or more message blocks, it may be unable temporarily to receive text into the buffer. This condition occurs when the current contents of the buffer are being transferred to an output device. When this happens, the 2770 sends a WACK sequence instead of the usual alternating acknowledgment.

Upon receiving the WACK, BTAM turns on bits 0 and 1 of DECFLAGS and posts a completion code of X'7F'. The user program should check DECFLAGS for this response, and if WACK was received, should send an ENQ character (as by a WRITE Inquiry macro). The ENQ should be sent regardless of whether the WACK was received in response (1) to text or (2) to the initial ENQ character (nonswitched point-to-point line) or ID ENQ sequence (switched point-to-point line). For a multipoint line, however, the ENQ should be sent only if the WACK was received in response to text. If it was received in response to initial selection, the user program should resend the selection characters, that is, reissue the WRITE macro. To determine whether the WACK was received in response to text or selection, examine the DECTPCOD field of the DECB. If it contains X'06', the WACK was received in response to selection. If it does not contain X'06', the WACK was received in response to text.

The user program should be arranged to keep responding to WACK sequences in this manner until the 2770 responds normally or until the user program wishes to abandon communication with the 2770 for the time being. In the latter case, the user program should issue the appropriate macro to break off transmission.

----

[1]If the source of the incoming data is an IBM 50 Magnetic Data Inscriber cartridge, it may be desirable for the user program to send NAK characters until a total of about 60 seconds has elapsed from the time the STX ENQ sequence was received. The MDI cartridge requires 45 seconds to rewind, and several seconds are required for the terminal operator to mount a new cartridge so that transmission can resume.

2770 Wishes to Transmit (RVI)

At the time the 2770 receives a selection sequence from the central computer (multipoint line only), it may signal the central computer that it wishes to transmit instead of receive. To do this, it sends an RVI sequence instead of an ACK-0 (the normal response to selection) BTAM accepts the RVI in lieu of the ACK-0, turns on bits 1 and 6 of DECFLAGS, and posts a completion code of X'7F'. The user program should check DECFLAGS for this response and proceed as follows.

When the RVI is received in response to selection, the program should issue a READ Initial macro if it wishes to allow the 2770 to transmit.

TERMINAL FUNCTION CONTROL

There are six characters in each code (EBCDIC and USASCII) that control terminal functions. These are sometimes referred to as end-to-end control characters (as distinguished from data link control characters).

- EM (end-of-medium)
  This character is used to indicate the end of data on paper or magnetic tape. It is transmitted as data and reproduced in paper tape at the 1018 paper tape punch. (It is not sent to the IBM 50 Magnetic Data Inscriber, as this is an input device only.)

- IRS (Interchange Record Separator) (EBCDIC)
  RS (Record Separator) (USASCII)
  This character is used to indicate the end of data in a punched card. When the contents of a card are read into the buffer, the control unit inserts an IRS (RS) character into the buffer following the last data character read from the card. If the contents of a buffer are sent to the paper tape punch, the IRS (RS) characters are also punched in the tape, so that cards can be punched from the tape. When sending data from the buffer to the card punch or printer, each IRS (RS) character encountered in data causes the control unit to command the card punch to eject a card, or the printer to perform the new line function.

- NL (New Line)
  The NL character defines a print line when data is to be printed. If data containing NL characters is sent to a card punch or paper tape punch, the NL characters are punched.

- DC1
  DC2 (Device Control)
  DC3

  These characters are used to activate specific devices attached to the 2772 control unit. Their use is explained under Terminal Polling and Selection.

- ESC (Escape)
  This character and a defined graphic character that follows it are called an escape sequence. Escape sequences are used to control formatting of data on output devices, as explained under 2213 Printer and 2265 Model 2 Display Station.

- VT (Vertical Tab)
  FF (Forms Feed)
  These two characters are used to control formatting on the 2213 printer, as explained under 2213 Printer.

Placing terminal function control characters in message text is not a BTAM function; they must be placed there by the terminal operator, programmer, or preparer of input media (e.g., cards, tape).

2213 Printer

Vertical forms control for the printer may be regulated by a carriage control tape contained within the printer, or by control commands consisting of escape sequences (ESC followed by a defined character). An escape sequence specifies the number of line spaces to be skipped following printing of the line in which the escape sequence appears, or specifies the channel number of the carriage control tape that is to govern forms motion. The escape sequence must be contained in the first two positions of a record sent to the printer. These two characters are not printed on the forms. The escape sequences and their corresponding functions are given in Figure 29.

Vertical forms control may also be actuated by the Vertical Tab (VT) character, which causes skip-to-channel-2 of the carriage control tape and the Forms Feed (FF) character, which causes skip-to-channel-1. These two characters differ in effect from the escape sequences in that forms motion takes place immediately upon detection of the VT or FF character, whereas forms motion caused by detection of an escape sequence does not occur until the entire line containing the escape sequence has been printed.

2265 Model 2 Display Station

Two 2265 Model 2 Display Station control functions are activated by two-character escape sequences contained within the message data sent to the display station via a Write operation. These functions and their associated escape sequences are:

Erase/Write (ESC U)

Write at Line Address (ESC ').

The escape sequence must be the first two characters following the STX character that begins a message or message block. Both the ESC U and ESC ' sequences may be contained in a single message (though not in the same block).

Erase/Write: To erase the screen of a display station the station must be selected and the program must send:

STX ESC U (text) ETX or

STX ESC U (text) ETB.

The screen is erased, the cursor is positioned at the first available display position of the screen (upper left corner), and the data represented by (text) is displayed.

Write at Line Address: This control function allows the program to select a specific line where the data containing the escape sequence is to be displayed. The program must send:

STX ESC ' x (text) ETX or

STX ESC ' x (text) ETB

where x represents the line address. The line address is a hexadecimal code specifying the display line where the message data is to begin. Display line numbers and corresponding line addresses are shown in Figure 30.

EBCDIC Sequence	USASCII Sequence	Forms Motion After Printing (or)	Skip to Carriage Control Tape Channel
ESC /	ESC Q	Single space	
ESC S	ESC R	Double space	
ESC T	ESC S	Triple space	
ESC A	ESC A		1
ESC B	ESC B		2
ESC C	ESC C		3
ESC D	ESC D		4
ESC E	ESC E		5
ESC F	ESC F		6
ESC G	ESC G		7
ESC H	ESC H		8
ESC I	ESC I		9
ESC J	ESC J		10
ESC K	ESC K		11
ESC L	ESC L		12
ESC M	ESC M	Space suppress	

Figure 29. IBM 2213 Vertical Forms Control Escape Sequences

Conversational Mode

This special feature enables the 2772 to accept a text response to an inquiry without having to be selected before receiving the response. With this feature, the user may include in his BTAM program the coding required to initiate a Write Continue operation to the 2770 immediately following the last block of data received from the same 2770 on a Read operation. This Write Continue operation may be followed by other Write Continue operations to the same 2770. To read more data from the 2770, the user must issue another READ Initial macro to poll the input unit again.

FIELD-CONTROL OPERATION

This special feature permits operator or program entry of three field modifiers that can be entered individually in any given character location in the display buffer.

Function	Escape Sequence	
Erase screen	ESC U	
Erase screen and display message	ESC U (text)	
Write at Line Address	ESC ' x (text)	
Display Line Number	Address Code	
	15 lines	12 lines
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	A	A
11	B	B
12	C	C
13	D	
14	E	
15	F	

Figure 30. IBM 2265 Erase and Write-at-Line Address Control Characters

Protected Data

Two of the three field modifiers provided by the Field-Control Operation special feature serve to identify the beginning and end of a field of data that is to be protected. They are:

- Protected-Data-Field Modifier (ESC Z): The presence of this modifier in the display buffer identifies the start of a field of protected data and prevents manual erase or over-write of the data.
- End-Field Modifier (ESC 9): The presence of this modifier in the display buffer identifies the end of a protected-data field and automatically terminates the field-control operation.

These field-control modifiers are entered into the system via escape (ESC) sequences. The characters Z and 9 are stored in the display buffer and they are reproduced on the display screen as:

Protected Data Field Modifier -- (
End-Field Modifier --)

When the (and) symbols are not preceded by ESC, they may be used as normal data. During a Read or Write operation, if the display cursor encounters a Protected-Data-Field Modifier it moves over the field until it reaches the End-Field Modifier. The cursor then locates in the next display position beyond the End-Field Modifier and normal Read or Write operation continues.

Tab Set

The presence of the Tab Set Character Field Modifier in the display buffer identifies the position as a Tab Set character location. When a Horizontal Tab (HT) is received from the central computer or from the keyboard, the HT character is stored in the first unprotected character space, and then the cursor automatically advances to one character space beyond the next Tab Set character.

The Tab Set Character Field Modifiers are entered into the system via a format message. The first data in this message should be ESC HT, to set up a tab-set sequence. Each HT following this represents a Tab Set character. A vertical bar is displayed in each line from and including the line containing the cursor to the bottom of the screen for each Tab Set character. This character cannot be written within a protected field. Caution must be exercised to see that the cursor is not positioned in a protected field when a Tab Set character is to be written. The tab-set sequence is not terminated until the New Line (NL) character is entered.

Once this format message has been stored, messages may be transmitted without spaces, as they need contain only the Horizontal Tab (HT) character to provide formatting. The HT character is stored and will be read back to provide printer formatting and/or better communications line efficiency. If an HT is sent and there are no Tab Field Modifiers, the cursor is positioned at the beginning of the next line.

RECORD FORMATS

The basic 2772 control unit has two 128-byte buffers. As a special feature the 2772 can have buffers of 256 bytes each. The basic 2772 can send or receive one message block per transmission. Thus, in non-transparent mode, messages appear on the line in the format:

STX	...text...	ETB (or ETX)

In transmission to the 2772, only the text portion of the message is transferred to the output device. The definition of text depends on the type of device.

For the paper tape punch, text consists of the data characters and any end-to-end control characters present in the data. For the card punch, any escape sequences, NL and EM characters are considered text and do not cause ejecting of a card from the punch; the IRS (RS), ETB, and ETX do cause card ejection. For the printer, the NL and IRS (RS) characters and escape sequences are not considered part of text.

In nontransparent mode, the maximum record length is 128 characters for the basic 2772 and 256 characters for the 2772 with the Expanded Buffer feature. The STX, ETB, ETX, and the device control characters (DCx) do not go into the buffer. Records exceeding the buffer size cause an I/O buffer to overrun error, which causes the 2772 to send a NAK in response to the received block. BTAM will retry the Write operation seven times, then post the operation complete with error.

Each IRS (RS) character in data transferred from the buffer to the card punch causes the card currently being punched to be ejected and a new card fed. This action also occurs if no IRS (RS) character has been detected by the time 80 consecutive data characters have been sent to the punch.

Printing: Data to be sent to the printer may be formatted into print lines of 132 characters or less by the use of IRS (RS) or NL characters. If neither of these characters is detected by the time 132 characters have been sent to the printer, successive data is printed on the next line. The new line function also occurs if the printer reaches a tab stop.

Display: Records exceeding the length of the display line are not truncated, but are continued on the next display line. In nontransparent mode, variable length records may be sent to the 2772. The number of records per transmission is not restricted except by buffer size. The STX, ETB, ETX, and DCx characters do not enter the buffer. All other characters, including escape sequences and end-to-end control characters, occupy positions in the buffer. In transparent mode, variable length blocks may be sent to the 2772. A block consists of one record, since end-to-end controls are not recognized in transparent mode. The length of the block may not exceed the buffer size.

Transmission of Blank Cards

Basic 2772: In either transparent or nontransparent mode, blank cards are read into the buffer and transmitted just as are cards containing data.

2772 with Expanded Buffer Feature: In nontransparent mode, data from the card reader is packed. That is, each card is read into the buffer, then scanned from column 80 backward until a data character is reached. The control unit then inserts an IRS (RS) character in the buffer at the next position. The next card is read into the buffer beginning at the next following position. Thus, card definition is maintained while unnecessary blanks at the end of the card are deleted. For this reason, the 2772, when equipped with the Expanded Buffer feature, does not transmit blank cards, in nontransparent mode. In transparent mode, data is not packed in the manner indicated above, and blank cards are transmitted.

Communication between the central computer and the station control unit of an IBM 2972 General Banking Terminal system employs an eight-bit transmission code and BSC multipoint data link control procedures. The makeup of the character set is as follows.

BSC Data Link Control Characters

The transmission code bit patterns for the data link control characters are identical to the EBCDIC bit patterns for the same characters. The 2972 station control unit can send and can receive and respond functionally to, these data link control characters and sequences:

Character	Bit Pattern (Hex)
STX	02
ETX	03
DLE	10
ETB	26
ENQ	2D
SYN	32
EOT	37
NAK	3D
ACK-0	1070
ACK-1	1061
WACK	106B
RVI	107C

The 2972 does not send the SOH (X'01') and ITB (X'1F') characters. However, it can receive them, but does not respond functionally to them. (This provides compatibility with other types of remote BSC stations that may be attached to the same multipoint line.)

Graphic and Terminal Function Control Characters

Graphic characters are the alphabetic and numeric characters and the special symbols that can be printed on the 2980 teller and administrative stations, or that these stations can send to the central computer. HT (horizontal tab), NL (new line), and Passbook Index are examples of terminal function control characters. The 2972 station control unit passes graphic and control characters between the communications line and the 2980s connected to the station control unit.

The character sets for the different models of the 2980 vary in the specific characters they include and in the individual transmission code bit patterns that represent the characters. For example, the bit pattern X'D3' represents a 6, in numeric shift, and L, in alphabetic shift, for the 2980 Model 1. For the Model 4, however, the same bit pattern, X'D3', represents L, in numeric shift, and Q, in alphabetic shift.

BTAM does not provide translation tables for user-program translation between EBCDIC and transmission codes. Appendix K shows the correspondence between each transmission code bit pattern and the characters that bit pattern represents, for each of the models (1, 2, and 4) of the 2980 stations.

IBM 3735 PROGRAMMABLE BUFFERED TERMINAL--PROGRAMMING CONSIDERATIONS

The 3735 Programmable Buffered Terminal is a stand-alone programmable terminal. The 3735 contains a communication interface and the controls necessary to use the BTAM BSC facilities to transmit properly assembled and structured Form Description programs to any terminal in the network, and to receive messages and data from the 3735 terminals.

Detailed information on the facilities that the 3735 Programmable Buffered Terminal provide is found in the 3735 Programmer's Guide, GC30-3001.

The Programmer's Guide describes the methods and facilities necessary to design, write, and generate form description programs.

The control units, display stations, and printers that make up the IBM 3270 Information Display System are supported by BTAM under control of the MFT or MVT options of the System/360 Operating System. Support for local or remote 3270 display systems or both can be included in BTAM.

3270 DEVICES SUPPORTED

BTAM supports the following remote 3270 control unit and devices:

- 3271 control unit, models 1 and 2
- 3277 display station, models 1 and 2
- 3275 display station, models 1 and 2
- 3284 printer, models 1, 2, and 3
- 3286 printer, models 1 and 2

The 3271 control unit must be attached to either a 2701 Data Adapter Unit or a 2703 Transmission Control Unit.

BTAM supports the following local 3270 control unit and devices:

- 2372 control unit, models 1 and 2
- 3277 display station, models 1 and 2
- 3284 printer, models 1 and 2
- 3286 printer, models 1 and 2

The 3272 control unit must be attached to a selector, multiplexer, or block multiplexer channel.

For more information about remote and local 3270 configurations, see *IBM 3270 Information Display System, Component Description*, GA27-2749.

3270 CAPABILITIES SUPPORTED

BTAM supports the following remote and local 3270 capabilities:

- Read modified fields from device buffer
- Write to device buffer
- Erase and write to device buffer
- Erase all unprotected fields in device buffer
- Read modified fields from device buffer from position
- Read from device buffer
- Read from device buffer from position

In addition, BTAM supports the remote 3270 capability of copying from the buffer of one remote device into the buffer of another remote device on the same control unit.

For remote 3270 display systems, capabilities are used through a combination of BTAM READ and WRITE macro instructions for nonswitched multipoint BSC stations and data link and end-to-end control characters in output messages. For local 3270 display systems, capabilities are used through local types of BTAM READ and WRITE macro instructions (sepcified by means of the optype operand).

For more information about remote and local 3270 capabilities, see *IBM 3270 Information Display System, Component Description*, GA27-2749.

REMOTE 3270 DISPLAY SYSTEM

Remote 3270 display stations and printers are supported by BTAM as BSC stations connected to nonswitched multipoint lines using either EBCDIC or ASCII transmission code.

DEFINING COMMUNICATIONS LINE GROUPS

See "Defining Communications Line Groups" in the general section "Defining the Teleprocessing System," and see Appendix D.

The UNIT operand of the IODEVICE system generation macro instruction must specify BSC3 for the remote 3270 display system. The DCB macro instruction operands that apply to the remote 3270 display system are: DSORG, MACRF, DDNAME, BUFNO, BUFL, BUFCB, EXLST, BFTEK, LERB, EROPT, DEVD, MODE, and CODE.

DEFINING AND MODIFYING TERMINAL LISTS

See "Defining and Modifying Terminal Lists" in the general section "Defining the Teleprocessing System," see "Defining Terminal Lists" under the heading "BSC Nonswitched Multipoint Operation" in the section "BSC Read and Write Operations," and see Appendix A.

Defining Terminal Lists

The DFTRMLST macro instruction is used to define terminal lists for the remote 3270 display system.

Each control unit has a one-character polling address (see Figure 3270-1) and a one-character selection address (see Figure 3270-2). Each display station or printer has its own one-character address for specific polling and selection (see Figure 3270-3), and all devices share a one-character address for general polling (see Figure 3270-3). Double addressing is used for both control unit and device. Each five-character polling or selection sequence has the format:

XX	XX	YY	YY	ENQ

where XX is the hexadecimal representation in EBCDIC or ASCII of the control unit address for polling or selection, YY is the hexadecimal representation of the device address, and ENQ is X'2D' for EBCDIC or X'05' for ASCII. In the polling list, the last entry must be five EOT characters (X'37' for EBCDIC or X'04' for ASCII).

Control Unit Number	Address Character	EBCDIC Hexadecimal Representation	ASCII Hexadecimal Representation
0	SP	40	20
1	A	C1	41
2	B	C2	42
3	C	C3	43
4	D	C4	44
5	E	C5	45
6	F	C6	46
7	G	C7	47
8	H	C8	48
9	I	C9	49
10	¢ ([)	4A	5B
11	.	4B	2E
12	<	4C	3C
13	(	4D	28
14	+	4E	2B
15	|	4F	21
16	&	50	26
17	J	D1	4A
18	K	D2	4B
19	L	D3	4C
20	M	D4	4D
21	N	D5	4E
22	O	D6	4F
23	P	D7	50
24	Q	D8	51
25	R	D9	52
26	! (])	5A	5D
27	$	5B	24
28	*	5C	2A
29	)	5D	29
30	;	5E	3B
31	¬	5F	5E

Figure 3270-1. Control Unit Addresses for Polling List Entries and for Identification in Input Messages (with hexadecimal representations in EBCDIC and ASCII)

Control Unit Number	Address Character	EBCDIC Hexadecimal Representation	ASCII Hexadecimal Representation
0	-	60	2D
1	/	61	2F
2	S	E2	53
3	T	E3	54
4	U	E4	55
5	V	E5	56
6	W	E6	57
7	X	E7	58
8	Y	E8	59
9	Z	E9	5C
10	(\)	6A	7C
11	,	6B	2C
12	%	6C	25
13		6D	5F
14	>	6E	3E
15	?	6F	3F
16	0	F0	30
17	1	F1	31
18	2	F2	32
19	3	F3	33
20	4	F4	34
21	5	F5	35
22	6	F6	36
23	7	F7	37
24	8	F8	38
25	9	F9	39
26	:	7A	3A
27	#	7B	23
28	@	7C	40
29	'	7D	27
30	=	7E	3D
31	"	7F	22

Figure 3270-2. Control Unit Addresses for Selection List Entries (with hexadecimal representations in EBCDIC and ASCII)

Device Number	Address Character	EBCDIC Hexadecimal Representation	ASCII Hexadecimal Representation
0	SP	40	20
1	A	C1	41
2	B	C2	42
3	C	C3	43
4	D	C4	44
5	E	C5	45
6	F	C6	46
7	G	C7	47
8	H	C8	48
9	I	C9	49
10	¢ ([)	4A	5B
11	.	4B	2E
12	<	4C	3C
13	(	4D	28
14	+	4E	2B
15	|	4F	21
16	&	50	26
17	J	D1	4A
18	K	D2	4B
19	L	D3	4C
20	M	D4	4D
21	N	D5	4E
22	O	D6	4F
23	P	D7	50
24	Q	D8	51
25	R	D9	52
26	! (])	5A	5D
27	$	5B	24
28	*	5C	2A
29	)	5D	29
30	;	5E	3B
31	¬	5F	5E
General Poll	"	7F	22

Figure 3270-3. Device Addresses for Polling and Selection List Entries and for Identification in Input Messages (with hexadecimal representations in EBCDIC and ASCII)

Examples: To define an open polling list for devices 1, 2, and 3 on control unit 7 using EBCDIC transmission code:

```
OPLIST  DFTRMLST   AUTOLST,(C7C7C1C12D,
                   C7C7C2C22D,C7C7C3C32D,
                   3737373737)
```

To define a wraparound polling list for devices 1, 2, and 3 on control unit 7 and a general poll on control unit 8 using ASCII:

```
WPLIST  DFTRMLST   AUTOWLST,(4747414141O5,
                   4747424205,4747434305,
                   4848222205,0404040404)
```

To define a selection list for devices 1, 2, and 3 on control unit 7 using EBCDIC:

```
SLIST   DFTRMLST     OPENLST,(E7E7C1C12D,
        E7E7C2C2D,E7E7C3C32D)
```

Modifying Terminal Lists

The CHGNTRY macro instruction is used to modify terminal lists for the remote 3270 display system. (if wraparound polling is being done, the RESETPL macro instruction is used first to terminate polling.)

Example: To suspend the poll on device 3 (from the first example under "Defining Terminal Lists"):

```
SPOLL CHGNTRY   OPLIST,AUTOLST,3,5,,SKIP
```

BUFFER MANAGEMENT

See the general section "Buffer Management." Programmer buffering or dynamic buffering can be used for the remote 3270 display system.

CODE TRANSLATION

See the general section "Code Translation." and see Appendix E.

To allow remote 3270 messages to be translated between EBCDIC and ASCII using the BTAM RASA and SASA translation tables, in the I/O interface code for six-bit structured data in all 3270 messages, the setting of the two high-order bits is determined by the setting of the six low-order bits in the byte (see Figure 3270-4). Six-bit structured data includes the WCC and CCC, attribute character, cursor and buffer addresses, remote control unit address, remote device address, and sense and status bytes; for more information, see IBM 3270 Information Display System, Component Description, GA27-2749.

00 0000 40 SP	01 0000 50 &	10 0000 60 -	11 0000 F0 0
00 0001 C1 A	01 0001 D1 J	10 0001 61 /	11 0001 F1 1
00 0010 C2 B	01 0010 D2 K	10 0010 E2 S	11 0010 F2 2
00 0011 C3 C	01 0011 D3 L	10 0011 E3 T	11 0011 F3 3
00 0100 C4 D	01 0100 D4 M	10 0100 E4 U	11 0100 F4 4
00 0101 C5 E	01 0101 D5 N	10 0101 E5 V	11 0101 F5 5
00 0110 C6 F	01 0110 D6 O	10 0110 E6 W	11 0110 F6 6
00 0111 C7 G	01 0111 D7 P	10 0111 E7 X	11 0111 F7 7
00 1000 C8 H	01 1000 D8 Q	10 1000 E8 Y	11 1000 F8 8
00 1001 C9 I	01 1001 D9 R	10 1001 E9 Z	11 1001 F9 9
00 1010 4A ¢	01 1010 5A !	10 1010 6A ¦	11 1010 7A :
00 1011 4B .	01 1011 5B $	10 1011 6B ,	11 1011 7B #
00 1100 4C <	01 1100 5C *	10 1100 6C %	11 1100 7C @
00 1101 4D (	01 1101 5D)	10 1101 6D _	11 1101 7D '
00 1110 4E +	01 1110 5E ;	10 1110 6E >	11 1110 7E =
00 1111 4F |	01 1111 5F ¬	10 1111 6F ?	11 1111 7F "

EBCDIC Bits 23 4567

EBCDIC Hex

Graphic Character

Note: The I/O interface code is obtained by overlaying columns 4,5,6, and 7 of standard EBCDIC code on columns C,D,E, and F.

Figure 3270-4. I/O Interface Code for Six-Bit Structured Data

(1) ASCII message (in hexadecimal)	02	47	41	27	44	48	11	44	20	4A	2E	20	53	4D	49	54	48	03
(2) Message characters	STX	G	A	'	D	H	DC1	D	Space	J	.	Space	S	M	I	T	H	ETX
(3) EBCDIC translation (in hexadecimal)	02	C7	C1	7D	C4	C8	11	C4	40	D1	4B	40	E2	D4	C9	E3	C8	03
(4) Message content	start of text	control unit: 7	device: 1	AID: ENTER key	cursor address: 0264		SBA order	buffer address: 0256		message text: J. SMITH								end of text

Figure 3270-5. Sample Input Message (showing translation from ASCII to EBCDIC)

(1) Message content	start of text	escape command: 1		WCC	SBA order	buffer address: 0064		SF order	attribute byte	message text: ENTER					end of text
(2) EBCDIC message (in hexadecimal)	02	27	F1	C3	11	C1	40	1D	60	C5	D5	E3	C5	D9	03
(3) Message characters	STX	ESC	1	C	DC1	A	Space	IGS / GS	-	E	N	T	E	R	ETX
(4) ASCII translation (in hexadecimal)	02	1B	31	43	11	41	20	1D	2D	45	4E	54	45	52	03

Figure 3270-6. Sample Output Message (showing translation from EBCDIC to ASCII)

Examples: Line 1 of Figure 3270-5 represents a message received from a display station using ASCII transmission code. (If the control unit and device addresses are to be checked against the entry in the polling list, this should be done before translation, since the terminal list entries are in transmission code.) Line 2 of Figure 3270-5 indicates the characters contained in the message. Line 3 shows the message after the TRNSLATE macro instruction has been used to translate from ASCII to EBCDIC. Line 4 indicates the content of the message for the problem program.

Line 1 of Figure 3270-6 indicates the content of a message from the problem program. Line 2 shows the message before the TRNSLATE macro instruction has been used to translate from EBCDIC to ASCII. Line 3 indicates the characters contained in the message. Line 4 represents the message to be sent to a display station using ASCII transmission code.

ACTIVATING AND DEACTIVATING THE TELEPROCESSING SYSTEM

See the general section "Activating and Deactivating the Teleprocessing System."

LINE CONTROL AND MESSAGE TRANSMISSION

See the general section "Line Control and Message Transmission," see "READ Macro Instructions" and WRITE Macro Instructions" under the heading "BSC Nonswitched Multipoint Operation" in the section "BSC Read and Write Operations," and see Appendixes B and G.

Read Operations

Read Modified Fields for Operator Input: To poll a device or control unit and, after some action by the display station operator, to read a message block, use the READ TI macro instruction. The polling list entry either specified by or in the polling list specified by the entry operand determines whether a single device or all devices on a control unit are polled.

- If the operator pressed the ENTER key, pressed a PF key, or selected a detectable field with the selector pen, the READ TI macro instruction causes a normal read. If the device buffer was formatted, the buffer specified by the inoutarea operand contains an index byte and a message block with the format:

STX	CU address	device address	AID	cursor address	SBA	buffer address	Text	SBA	buffer address	Text		ETB ETX
1	1	1	1	2	1	2		1	2			1

If the device buffer was unformatted, the buffer specified by the inoutarea operand contains an index byte and a message block with the format:

STX	CU address	device address	AID	cursor address	Text		ETB ETX
1	1	1	1	2			1

Data from remote 3270 devices is blocked with a nominal length of 256 bytes (including data link control characters). The actual length of a block can be calculated from the DECCOUNT field in the DECB.

Since a response may contain more than one block, use READ TT macro instructions to read blocks until an EOT is received. If a specific poll was used and the device buffer was formatted, subsequent message blocks have the format:

STX		SBA	buffer address	Text		ETB ETX
1		1	2			1

If a specific poll was used and the device buffer was unformatted, subsequent message blocks have the format:

STX	Text		ETB ETX
1			1

A general poll may result in messages from more than one device attached to the control unit polled. The last block of a message from one device ends with an ETX. If a general poll was used and the device buffer was formatted, a message block following a block ending with an ETB has the format:

STX		SBA	buffer address	Text		ETB ETX
1		1	2			1

The message block following a block ending with an ETX has the format:

STX	CU address	device address	AID	cursor address	SBA	buffer address	Text	SBA	buffer address	Text		ETB ETX
1	1	1	1	2	1	2		1	2			1

If a general poll was used and the device buffer was unformatted, a message block following a block ending with an ETB has the format:

STX	Text		ETB ETX
1			1

The message block following a block ending with an ETX has the format:

STX	CU address	device address	AID	cursor address	Text		ETB ETX
1	1	1	1	2			1

Note: A message block received in response to a READ TT macro instruction is not preceded by an index byte.

- If the operator pressed the CLEAR key or pressed a PA key, the READ TI macro instruction causes a short read. The buffer specified by the inoutarea operand contains:

index byte	STX	CU address	device address	AID	ETX
1	1	1	1	1	1

- If a card or cards were read by means of the operator indentification card reader, the buffer specified by the inoutarea operand contains:

index byte	STX	CU address	device address	AID	0-37 characters	EOR / EOI	LRC	ETX
1	1	1	1	1		1	1	1

- If a test request message was entered and EROPT=T was not specified in the DCB, the buffer specified by the inoutarea operand contains:

index byte	SOH	%	/	STX	Text	ETX
1	1	1	1	1		1

- If an error status message was read, the buffer specified by the inoutarea operand contains:

index byte	SOH	%	R	STX	CU address	device address	sense/status bytes	ETX
1	1	1	1	1	1	1	2	1

To terminate a read operation, issue a READ TRV macro instruction after receiving a message block ending with an ETX. If a permanent I/O error occurs, either issue a READ TP macro instruction to reread the block or issue a WRITE TR macro instruction to terminate the operation.

Examples of READ macro instructions for reading modified fields for operator input are:

```
READ  DECB1,TI,DCB1,INBUF1,256,PNTRY1,2
READ  DECB2,TT,DCB1,INBUF2,256,,2
```

<u>Read Modified Fields</u>: To select a device and, independently of action by the display station operator, to read from the device, use a WRITE TIV macro instruction with the entry operand specifying the selection entry of the device and the outarea operand specifying a buffer that contains (in EBCDIC or ASCII);

STX	ESC	6	ETX
1	1	1	1

Follow the WRITE macro instruction with READ TT macro instructions.

In the first message block, if the AID byte contains neither C'-' nor C'Y' or if the first byte is an SOH, the buffer specified by the inoutarea opearnd contains a message block with one of the formats described under "Read Modified Fields for Operator Input." Otherwise, the contents of the buffer are unpredictable; the message is probably nonexistent or incomplete.

Examples of WRITE and READ macro instructions for reading modified fields are:

```
WRITE  DECB1,TIV,DCB1,(INBUF1,OUTBUF),
       (256,4),SYNTRY2,3
READ   DECB2,TT,DCB1,INBUF1,256,,3
```

<u>Read Modified Fields from Position</u>: To select a device and read from the device, use a WRITE TI macro instruction with the entry operand specifying the selection entry of the device and the inoutarea operand specifying a buffer that contains (in EBCDIC or ASCII):

STX	ESC	1	WCC		SBA	buffer address	ETX
1	1	1	1		1	2	1

Any data stream valid for a write operation may be used, but the WCC should inhibit reset of modified data tags and the last buffer address should indicate where the read modified operation is to start.

Follow the WRITE TI macro instruction with a WRITE TTV macro instruction with the outarea operand specifying a buffer that contains (in EBCDIC or ASCII):

STX	ESC	6	ETX
1	1	1	1

Follow the WRITE TTV macro instruction with READ TT macro instructions.

Input message blocks are the same as those described under "Read Modified Fields."

Examples of WRITE and READ macro instructions for reading modified fields from position are:

```
WRITE  DECB1,TI,DCB1,OUTBF1,8,SNTRY2,1
WRITE  DECB2,TTV,DCB1,(INBUF1,OUTBUF2),
       (256,4),,1
READ   DECB3,TT,DCB1,INBUF1,256,,1
```

<u>Read Buffer</u>: To select a device and read from the device, use a WRITE TIV macro instruction with the entry operand specifying the selection entry of the device and the outarea operand specifying a buffer that contains (IN EBCDIC or ASCII):

STX	ESC	2	ETX
1	1	1	1

Follow the WRITE TIV macro instruction with READ TT macro instructions.

If the device buffer was formatted, the first message block in the buffer specified by the inoutarea operand has the format:

STX	CU address	device address	AID	cursor address		SF	attribute	text		ETB / ETX
1	1	1	1	2		1	1			1

If the device buffer was unformatted, the first message block in the buffer specified by the inoutarea operand has the format:

STX	CU address	device address	AID	cursor address	text	ETB / ETX
1	1	1	1	2		1

If the device buffer was formatted, subsequent message blocks have the format:

STX		SF	attribute	text		ETB / ETX
1		1	1			1

If the device buffer was unformatted, subsequent message blocks have the format:

STX	text	ETB / ETX
1		1

Examples of WRITE and READ macro instructions for reading a buffer are:

```
WRITE  DECB1,TIV,DCB1,(INBUF1,OUTBUF),
       (256,4),SYNTRY2,2
READ   DECB2,TT,DCB1,INBUF1,256,,2
```

<u>Read Buffer from Position</u>: To select a device and read from the device, use a WRITE TI macro instruction with the entry operand specifying the selection entry of the device and the inoutarea operand specifying a buffer that contains (in EBCDIC or ASCII):

STX	ESC	1	WCC		SBA	buffer address	ETX
1	1	1	1		1	2	1

Any data stream valid for a write operation may be used, but the WCC should inhibit reset of modified data tags (if their setting is wanted in the input message), and the last buffer address should indicate where the read buffer operation is to start.

Follow the WRITE TI macro instruction with a WRITE TTV macro instruction with the outarea operand specifying a buffer that contains (in EBCDIC or ASCII):

STX	ESC	2	ETX
1	1	1	1

Follow the WRITE TTV macro instruction with READ TT macro instructions.

Input message blocks are the same as those described under "Read Buffer."

Examples of WRITE and READ macro instructions for reading a buffer from position are:

```
WRITE  DECB1,TI,DCB1,OUTBF1,8,SNTRY3,3
WRITE  DECB2,TTV,DCB1,(INBUF1,OUTBF2),
       (256,4),,3
READ   DECB3,TT,DCB1,INBUF1,256,,3
```

Write Operations

Write Buffer: To select a device and write a message block or blocks, use one of the folowing sequences of WRITE macro instructions:

- WRITE TIR
- WRITE TI
WRITE TR
- WRITE TI
WRITE TTR
- WRITE TI
WRITE TT
.
.
.
WRITE TR
- WRITE TI
WRITE TT
.
.
.
WRITE TTR

with the entry operand of the WRITE TIR or WRITE TI macro instruction specifying the selection entry of the device and the inoutarea operand of the WRITE TIR, WRITE TI, WRITE TT, or WRITE TTR macro instruction specifying a buffer that contains (in EBCDIC or ASCII):

STX	ESC	1	WCC	orders and text	ETX
1	1	1	1		1

For information about the WCC and the orders and text that may follow it, see IBM 3270 Information Display System, Component Description, GA27-2749. An SBA order sequence should follow immediately after the WCC, so that the write operation can be retried if an error occurs.

Examples of WRITE macro instructions for writing a buffer are:

```
WRITE  DECB1,TIR,DCB1,OUTBUF,128,SNTRY1,2

WRITE  DECB1,TI,DCB1,OUTBUF,128,SNTRY1,2
WRITE  DECB2,TR,DCB1,,,,2

WRITE  DECB1,TI,DCB1,OUTBF1,256,SNTRY2,3
WRITE  DECB2,TTR,DCB1,OUTBF2,128,,3

WRITE  DECB1,TI,DCB1,OUTBF1,256,SNTRY3,2
WRITE  DECB2,TT,DCB1,OUTBF2,256,,2
WRITE  DECB3,TR,DCB1,,,,2

WRITE  DECB1,TI,DCB1,OUTBF1,256,SNTRY2,1
WRITE  DECB2,TT,DCB1,OUTBF2,256,,1
WRITE  DECB3,TTR,DCB1,OUTBF3,128,,1
```

Erase and Write Buffer: To select a device, clear its buffer to nulls (binary zeros), and write a message block or blocks, use one of the sequences of WRITE macro instructions listed under "Write Buffer" with the entry operand of the WRITE TIR or WRITE TI macro instruction sepcifying the selection entry of the device and the inoutarea operand of the WRITE TIR, WRITE TI, WRITE TT, or WRITE TTR macro instruction specifying a buffer that contains (in EBCDIC or ASCII):

STX	ESC	5	WCC	orders and text	ETX
1	1	1	1		1

Examples of WRITE macro instructions for erasing and writing a buffer are the same as those given under "Write Buffer."

Erase Unprotected Fields: To select a device and set all unprotected fields in its buffer to nulls (binary zeros), use the WRITE TIR macro instruction or the WRITE TI and WRITE TR macro instructions with the instructions with the entry operand of the WRITE TIR or WRITE TI macro instruction specifying the selection entry of the device and the inoutarea operand specifying a buffer that contains (in EBCDIC or ASCII):

STX	ESC	?	ETX
1	1	1	1

For more information about the results of this operation, see IBM 3270 Information Display System, Component Description, GA27-2749.

Examples of WRITE macro instructions for erasing all unprotected fields are:

```
WRITE  DECB1,TIR,DCB1,OUTBUF,4,SNTRY1,2

WRITE  DECB1,TI,DCB1,OUTBUF,4,SNTRY1,2
WRITE  DECB2,TR,DCB1,,,,2
```

<u>COPY</u>: To select a device and copy into its buffer the contents of the buffer of another device on the same control unit, use the WRITE TIR macro instruction or the WRITE TI and WRITE TR macro instructions with the entry operand of the WRITE TIR or WRITE TI macro instruction specifying the selection entry of the device and the inoutarea operand specifying a buffer that contains (in EBCDIC or ASCII):

STX	ESC	7	CCC	device address	ETX
1	1	1	1	1	1

For more information about the CCC and the device address that follows it, see <u>IBM 3270 Information Display System, Component Description</u>, GA27-2749.

Examples of WRITE macro instructions for copying into a buffer are:

```
WRITE  DECB1,TIR,DCB1,OUTBUF,6,SNTRY1,2

WRITE  DECB1,TI,DCB1,OUTBUF,6,SNTRY1,2
WRITE  DECB2,TR,DCB1,,,,2
```

PROGRAMMING NOTES

If the response to a READ TI, READ TT, WRITE TIV, or WRITE TTV macro instruction is a message block ending with an ETB, follow with a READ TT macro instruction, <u>not</u> a WRITE TT or WRITE TTV macro instruction.

If a WRITE TI macro instruction is used to erase unprotected fields, follow with a WRITE TR macro instruction, <u>not</u> a WRITE TT, WRITE TIV, or WRITE TTV macro instruction.

If a WRITE TI or WRITE TT macro instruction is used to start a printer, follow with a WRITE TR macro instruction, <u>not</u> a WRITE TT, WRITE TIV, or WRITE TTV macro instruction.

ERROR RECOVERY PROCEDURES AND ERROR RECORDING

See the general section "Error Recovery Procedures and Error Recording," and see Appendixes B and C.

Error Conditions

An error status message should be read from the remote 3270 device if:

- A WRITE TI, WRITE TIR, or WRITE TIV macro instruction receives an RVI sequence in response to selection (completion code is X'7F'; bits 1 and 6 are on in the DECFLAGS field of the DECB)
- A WRITE TI, WRITE TIR, or WRITE TIV macro instruction receives an EOT in response to text (completion code is X'41'; bit 1 is on in the DECFLAGS field; and EOT is in byte 1 of the DECRSPN field of the DECB)
- A WRITE TIV or READ TI macro instruction receives a text block ending with an ENQ (completion code is X'41'; bit 1 is on in the DECFLAGS field)

To receive the error status message, the problem program should issue a READ TI macro instruction using the polling entry of the device for which completion was posted. See Appendix C for a description of the error status message and suggested actions based on its contents.

Exceptional Conditions

If the completion code is X'7F' and bits 1 and 6 are on in the DECFLAGS field of the DECB, and error status message was received in response to a READ TI macro instruction (or in response to a READ TT macro instruction is a general poll was used).

If the completion code is X'7F' and bits 0 and 1 are on in the DECFLAGS fields of the DECB, a WACK was received in response to a WRITE TI or WRITE TT macro instruction. If the write operation started a printer, this is a normal completion. A WRITE TR macro instruction must follow to reset the line.

RETRY OPTIONS

See the BSC3 retry options in the general section "Suggested Retry Options for BSC Read and Write Operations."

ON-LINE TESTING

See "On-Line Testing for Binary Synchronous Communications Lines" in the general section "On-Line Testing."

To receive standard IBM maintenance for a remote 3270 display system, the on-line testing facility must be available.

LOCAL 3270 DISPLAY SYSTEM

The functions provided by BTAM for remote stations have been extended to support local 3270 display stations and printers as local devices using EBCDIC code.

DEFINING THE LOCAL 3270 DISPLAY SYSTEM

Identifying Local 3270 Devices

The control unit and devices in a local 3270 display system are identified as local during system generation. For the local 3270 display system, the UNIT parameter of the IOCONTRL system generation macro instruction should specify 3272, and the MODEL parameter should specify 1 or 2. the IODEVICE system generation macro instruction operands that apply to the local 3270 display system are: UNIT, ADDRESS, MODEL, and FEATURE. For more information about the IOCONTROL and IODEVICE system generation macro instructions, see Appendix D. For more information about models and features of the local 3270 display system, see _IBM 3270 Information Display System, Component Description_, GA27-2749.

Example: The following system generation macro instructions are used to identify the local 3270 display systems shown in Figure 3270-7:

```
IOCONTRL   UNIT=3272,ADDRESS=22, . . .
IOCONTRL   UNIT=3272,ADDRESS=23, . . .

IODEVICE   UNIT=3277,ADDRESS=221, . . .
IODEVICE   UNIT=3286,ADDRESS=222, . . .
IODEVICE   UNIT=3277,ADDRESS=225, . . .
IODEVICE   UNIT=3284,ADDRESS=227, . . .
IODEVICE   UNIT=3277,ADDRESS=22E, . . .
IODEVICE   UNIT=3277,ADDRESS=231, . . .
IODEVICE   UNIT=3286,ADDRESS=233, . . .
```

Grouping Local 3270 Devices

Local 3270 devices are grouped together in the same way as remote terminals.

Each local 3270 device is equivalent to a communication line. Each group of local 3270 devices is equivalent to a line group.

Local 3270 devices may be grouped during system generation by means of the UNITNAME macro instruction. Or line groups may be specified during program execution by means of the UNIT parameter of the DD statement.

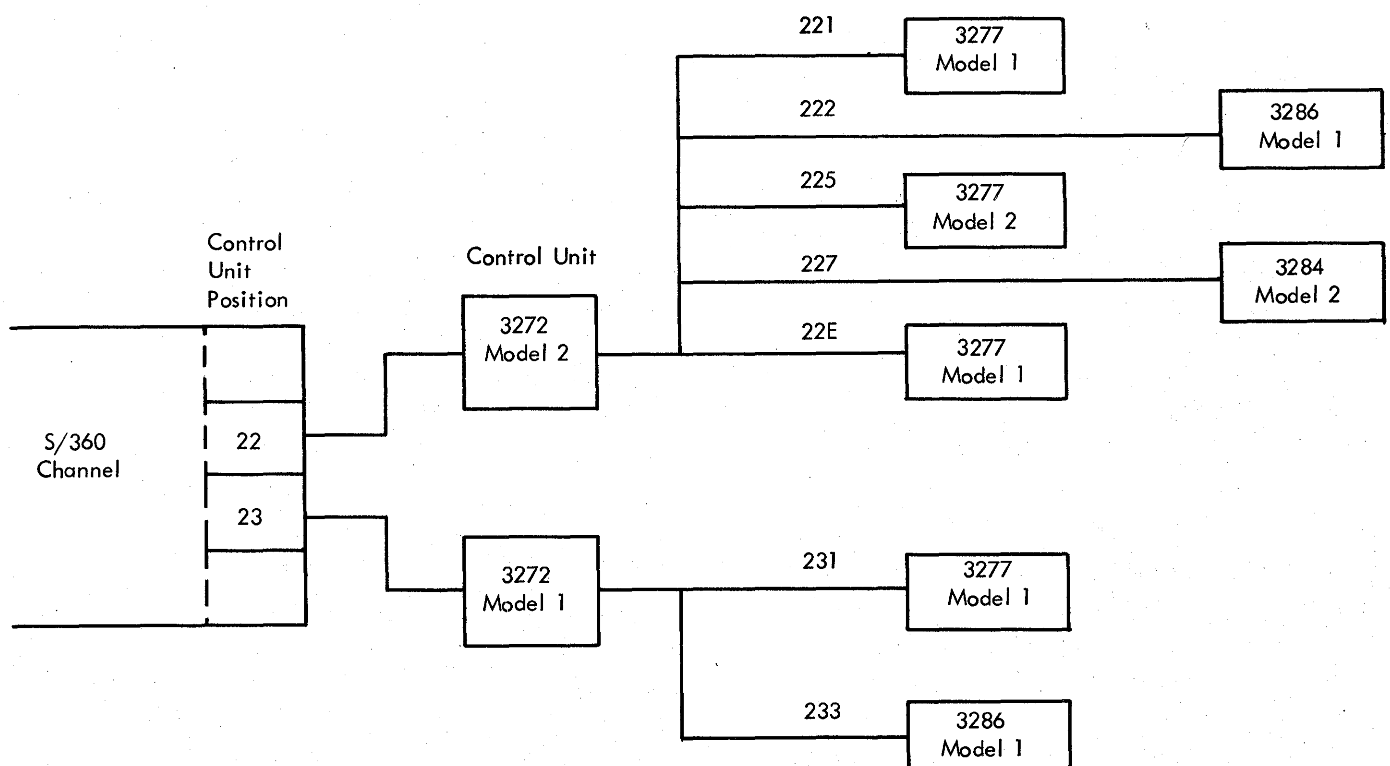


Figure 3270-7. Sample Local 3270 Display Systems Showing Device Addresses

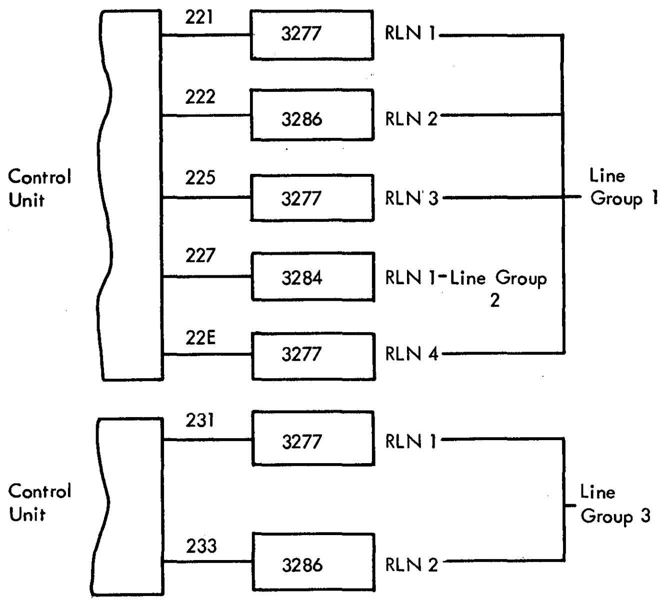


Figure 3270-8. Line Groups and Relative Line Numbers for Example 1.

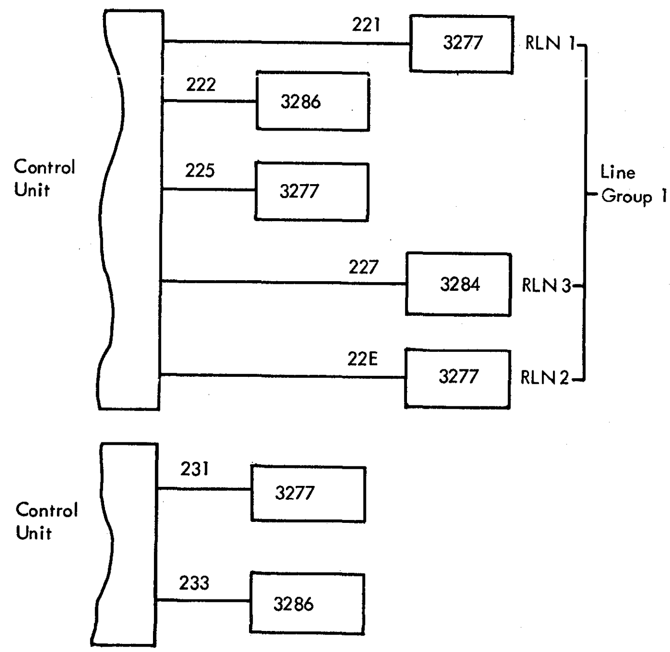


Figure 3270-9. Line Group and Relative Line Numbers for Example 2.

Local 3270 devices making up a line group are associated with one DCB. The DDNAME in the DCB must be the same as the name of the DD statement for the line group. A line group can contain up to 255 devices. A line group need not include all the devices on a control unit; it can include devices from several control units. Each local 3270 device is identified by its relative line number (RLN). The relative line numbers are determined by the order in which devices are grouped during system generation or program execution.

Example 1: To define (during system generation) the line groups and relative line numbers shown in Figure 3270-8 for the local 3270 display systems shown in Figure 3270-7, see "Example 1" under "Defining Communications Line Groups" in the general section "Defining the Teleprocessing System." (Address 221 would be coded in place of 021, 222 in place of 022, etc.)

Example 2: To define (during program execution) the line groups and relative line numbers shown in Figure 3270-9, see "Example 2" under "Defining Communications Line Groups" in the general section "Defining the Teleprocessing System." (Address 221 would be coded in place of 021, 227, in place of 027, etc.)

For a description of the DCB macro instruction, see "Defining Communications Line Groups" in the general section "Defining the Teleprocessing System." The DCB macro instruction operands that apply to the local 3270 display system are: DSORG, MACRF, DDNAME, BUFNO, BUFL, BUFCB, EXLST, and EROPT.

The local 3270 device can be identified in a READ or WRITE macro instruction by means of the DCB and the relative line number; no terminal list is needed. Therefore, the DFTRMLST macro instruction is not used for the local 3270 display system.

BUFFER MANAGEMENT

See the general section "Buffer Management." Dynamic buffering cannot be used for the local 3270 display system. The REQBUF and RELBUF macro instructions can be used for manipulating buffer pools.

CODE TRANSLATION

Since only EBCDIC code is used, code translation does not apply to the local 3270 display system.

In the I/O interface code for six-bit structured data in all 3270 messages, the setting of the two high-order bits is determined by the setting of the six low-order bits in the byte (see Figure 3270-4). Six-bit structured data includes the WCC, attribute character, and cursor and buffer addresses; for more information, see IBM 3270 Information Display System, Component Descripton, GA27-2749.

ACTIVATING AND DEACTIVATING THE LOCAL 3270 DISPLAY SYSTEM

See the general section "Activating and Deactivating the Teleprocessing System." The OPEN and CLOSE macro instructions are used to activate and deactivate line groups of local 3270 devices. The LOPEN macro instruction is used to activate a specific local 3270 device when the OPEN macro instruction has been unsuccessful.

ATTENTION INTERRUPTIONS AND READ INITIAL OPERATIONS

When the operator of a local 3270 display station carries out certain actions, such as pressing the ENTER or CLEAR key, pressing a PF of PA key, or selecting a detectable field with the selector pen, an I/O interruption, called an attention interruption, occurs. If a DCB has been opened for a line group that includes the display station, the attention interruption is recorded for the display station. If a DCB has not been opened, the attention interruption is ignored.

The READ TI macro instruction for the local 3270 display system includes the dc operand, which specifies a DCB, and the rln operand, which specifies a local 3270 display station in the line group associated with the DCB. When a READ TI macro instruction is issued, it causes a check (beginning with the display station specified by the rln operand) of whether an attention interruption has been recorded for any of the local 3270 display stations in the specified line group.

If an attention interruption has occurred, a read initial operation is started to read a message from the display station from which the attention interruption came. When the message has been received, the READ macro instruction is posted complete, and the relative line number of the display staion is placed into the DECPOLPT field of the DECB. Only one attention interruption is serviced for each READ TI macro instruction, and only one message is read. After a READ TI macro instruction has been issued, the problem program cannot issue another READ or WRITE macro instruction specifying the same DCB until either the read initial operation has been posted complete or the read request has been canceled by means of the RESETPL macro instruction. Attention interruptions that occur between read initial operations are recorded for the display stations and serviced by later READ TI macro instructions.

If no attention interruption has occurred for the display stations associated with the DCB (when a READ TI macro instruction is issued), a read request (that is, a pending read initial operation) is recorded for the line group. When an attention interruption comes from one of the display stations, the pending read initial operation is started for that display station, and the read request is cleared for the line group.

CHGNTRY Macro Instruction

A special form of the CHGNTRY macro instruction is used in handling attention interruptions from the local 3270 display station. A CHNGTRY macro instruction with the SKIP operand is used to have a display station skipped, so that a read initial operation is not started (that is, an EXCP macro instruction is not issued) for that display station when a READ TI macro instruction is issued, even though an attention interruption has occurred. (If, when a READ TI macro instruction is issued, all display stations in the line group are to be skipped, a pending read initial operation is recorded for the line group.) A CHGNTRY macro instruction with the ACTIVATE operand is used to have a display station activated, so that a read initial operation is started when a READ TI macro instruction is issued, and an attention interruption has occurred. If, when the display station is activated, a pending read initial operation has been recorded for the line group and an attention interruption has been recorded for that display station, the pending read initial operation is started (that is, and EXCP macro instruction is issued).

Note: This form of the CHGNTRY macro instruction affects only read initial operations; it does not affect other local 3270 read or write operations.

Name	Operation	Operand
[symbol]	CHGNTRY	dcbaddr, ATTLST, listposition,, {SKIP / ACTIVATE}

dcbaddr
specifies the address of the DCB associated with the line group that includes the local 3270 display station to be sikpped or activated.

ATTLST
specifies that the relative line numbers of the local 3270 display stations in the line group associated with the DCB are to be treated as an attention list.

listposition
specifies the relative line number of the local 3270 display station to be skipped or activated.

SKIP
specifies that the local 3270 display station is to be skipped; that is, an attention interruption that has been recorded for the display station is to be ignored if a READ TI macro instruction is issued.

ACTIVATE
specifies that the local 3270 display station is to be activated; that is, a read initial operation is to be started if an attention interruption has been recorded for the display station when a READ TI macro instruction is issued.

Return codes for this form of the CHGNTRY macro instruction are the same as those for the regular form (see "CHNGTRY Macro Instruction" in the general section "Defining the Teleprocessing System").

RESETPL Macro Instruction

A special form of the RESETPL macro instruction is used for the local 3270 display system to cancel a read initial operation that is pending (because a READ TI macro instruction was issued when an attention interruption has not occurred).

Name	Operation	Operand
[symbol]	RESETPL	decbaddr [,ATTENT]

decbaddr
specifies the address of the DECB for the pending read initial operation that is to be canceled.

ATTENT
specifies that only the instructions required to cancel a pending read initial operation for a local 3270 display system are to be generated by this macro instruction.

If no second operand is specified, instructions are generated to determine the line type, and the proper instructions for that line type are executed.

Note: If POLLING or ANSRING is specified as the second operand, instructions for the local 3270 display system are not generated.

Programming Notes: The RESETPL macro instruction does not halt read initial operations that have been started; it does prevent the outstanding READ TI macro instruction from having any subsequent attention interruptions serviced. If a read initial operation is pending (that is, it has not been started), the IOBs are marked free, and a completion code of X'48' is posted in the ECB. If a read initial operation has been started, the IOBs are not marked free, and a completion code is not posted in the ECB by the RESETPL macro instruction.

Follow the READ TI macro instruction with a WAIT macro instruction with the ECBLIST operand (or a TWAIT macro instruction) that specifies multiple ECBs, the ECB in the DECB specified by the READ TI macro instruction and another ECB. To allow the RESETPL macro instruction to be issued to cancel the read initial operation, post the other ECB. Provided that the RESETPL macro instruction gave a return code of X'00' or X'04', follow the RESETPL macro instruction with a WAIT macro instruction that specifies the ECB in the DECB specified by the READ TI macro instruction. A completion code of X'7F' indicates that a read initial operation had been started and that it completed sucessfully. Other READ and WRITE macro instructions can then be issued.

Return Codes: When this form of the RESETPL macro instruction is used, the return codes have the following meanings:

00 This code is set when the pending read initial operation for the specified DECB was canceled successfully.
04 This code is set when a read initial operation for the specified DECB was started and will complete normally (see "Programming Notes" above).
08 This code is set for an illegal request (that is, BTAM found that the specified DECB is not associated with a READ TI macro instruction for a local 3270 display system).
0C This code is set for an unsuccessful request (that is, BTAM found an invalid control block).
10 This code is set when there was no read initial operation (either pending or started) for the specified DECB.

READ AND WRITE OPERATIONS

See "Read and Write Macro Instructions" and TAble 6A in the general section "Line Control and Message Transmission," see

"READ Macro Instructions" and "WRITE Macro Instructions" under the heading "Local IBM 3270 Display System" in the section "Local Read and Write Operations," and see Appendixes B and G.

Read Operations

Read Modified Fields for Operator Input: To read a message from a display station after an attention interruption has come from it as the result of some action by the display station operator, use the READ TI macro instruction. The rln operand specifies which display station in the line group should be checked first for an attention interruption.

- If the operator pressed the ENTER key, pressed a PF key, or selected a detectable field with the selector pen, the READ TI macro instruction causes a normal read. If the device buffer was formatted, the buffer specified by the inoutarea operand contains a message with the format:

AID	cursor address	SBA	buffer address	text	SBA	buffer address	text
1	2	1	2		1	2	

  If the device buffer was unformatted, the buffer specified by the inoutarea operand contains a message with the format:

AID	cursor address	text
1	2	

  The length of the message can be calculated from the DECCOUNT field in the DECB, which contains the residual count. If the residual count is zero, the READ TMP macro instruction may be used to continue reading data.

- If the operator pressed the CLEAR key or pressed a PA key or if a card was extracted from the badge reader, the READ TI macro instruction causes a short read. The buffer specified by the inoutarea operand contains:

AID
1

- If a card or cards were read by means of the operator identification card reader, the buffer specified by the inoutarea operand contains:

AID	0-37 characters	EOR / EOI	LRC
1		1	1

- If a test request message was entered and EROPT=T was not specified in the DCB, the buffer specified by the inoutarea operand contains:

SOH	%	/	STX	text
1	1	1	1	

The relative line number of the display station from which the message was read is placed into the DECPOLPT field of the DECB.

To cancel the read initial operation requested by a READ TI macro instruction (if the operation is pending), issue the RESETPL macro instruction specifying the DECB from the READ TI macro instruction.

Do not issue a READ TI macro instruction specifying a DCB associated with a line group that contains only printers, since the local 3270 printer cannot generate attention interruptions.

An example of a READ macro instruction for reading modified fields for operator input is:

```
READ  DECB1,TI,DCB1,INBUF1,256,,2
```

Read Modified Fields: To read a message from a device independently of action by the display station operator, use the READ TM macro instruction.

If the AID byte in the message contains neither C'-' nor C'Y' or if the first byte is an SOH, the buffer specified by the inoutarea operand contains a message with one of the formats described under "Read Modified Fields for Operator Input." Otherwise, the contents of the buffer are unpredictable; the message is probably nonexistent or incomplete.

The length of the message can be calculated from the DECCOUNT field in the DECB, which contains the residual count. If the residual count is zero, the READ TMP macro instruction may be used to continue reading data. The relative line number of the device from which the message was read is placed into the DECPOLPT field of the DECB.

An example of a READ macro instruction for reading modified fields is:

```
READ  DECB1,TM,DCB1,INBUF1,256,,3
```

Read Modified Fields from Position: To read a message from a device, use the READ TMP macro instruction with the entry operand specifying an area that contains:

WCC	SBA	buffer address
1	1	2

The WCC should inhibit reset of modified data tags, and the buffer address should indicate where the read modified operation is to start. Data transfer begins with the first modified field at or following the buffer address specified.

Input messages are the same as those described under "Read Modified Fields."

An example of a READ macro instruction for reading modified fields from position is:

```
READ  DECB1,TMP,DCB1,INBUF1,256,ENTRY1,1
```

Read Buffer: To read a message from a device, use the READ TB macro instruction. If the device buffer was formatted, the buffer specified by the inoutarea operand contains a message with the format:

AID	cursor address		SF	attribute	text	
1	2		1	1		

If the device buffer was unformatted, the buffer specified by the inoutarea operand contains a message with the format:

AID	cursor address	text
1	2	

The length of the message can be calculated from the DECCOUNT field in the DECB, which contains the residual count. If the residual count is zero, the READ TBP macro instruction may be used to continue reading data. The relative line number the device from which the message was read is placed into the DECPOLPT field of the DECB.

An example of a READ macro instruction for reading a buffer is:

```
READ  DECB1,TB,DCB1,INBUF1,256,,2
```

Read Buffer from Position: To read a message from a device, use the READ TBP macro instruction with the entry operand specifying an area that contains:

WCC	SBA	buffer address
1	1	1

The WCC should inhibit reset of modified data tags (if their setting is wanted in the input message), and the buffer address should indicate where the read buffer operation is to start.

Input messages are the same as those described under "Read Buffer."

An example of a READ macro instruction for reading a buffer from position is:

```
READ  DECB1,TBP,DCB1,INBUF1,256,ENTRY2,3
```

Write Operations

Write Buffer: To write a message to a device, use the WRITE TI macro instruction with the inoutarea operand specifying a buffer that contains:

WCC	orders and text
1	

For information about the WCC and the orders and text that may follow it, see IBM 3270 Information Display System, Component Description, GA27-2749. An SBA order sequence should follow immediately after the WCC, so that the write operation can be retried if an error occurs.

An example of a WRITE macro instruction for writing a buffer is:

```
WRITE  DECB1,TI,DCB1,OUTBUF,256,,2
```

Erase-and-Write Buffer: To clear its buffer to nulls (binary zeros) and write a message to a device, use the WRITE TS macro instruction with the inoutarea specifying a buffer that has the same contents as described under "Write Buffer."

An example of a WRITE macro instruction for erasing and writing a buffer is:

```
WRITE  DECB1,TS,DCB1,OUTBUF,256,,2
```

Erase Unprotected Fields: To set all unprotected fields in the buffer of a device to nulls (binary zeros), use the WRITE TUS macro instruction with the inoutarea operand specifying any real address and the inoutlength operand specified as one.

For more information about the results of this operation, see IBM 3270 Information Display System, Component Description, GA27-2749.

An example of a WRITE macro instruction for erasing all unprotected fields is:

```
WRITE  DECB1,TUS,DCB1,OUTBUF,1,,2
```

ERROR RECOVERY PROCEDURES AND ERROR RECORDING

See the general section "Error Recovery Procedures and Error Recording," and see Appendix B.

RETRY OPTIONS

See the local 3270 retry options in the general section "Suggested Retry Options for Local Read and Write Operations."

ON-LINE TESTING

See "On-Line Testing for Local 3270 Display System" in the general section "On-Line Testing."

To receive standard IBM maintenance for a local 3270 display system, the on-line testing facility must be available.

SYSTEM GENERATION

See Appendix D for information about the operands that must be included in the IOCONTRL and IODEVICE system generation macro instructions when generating an operating system that includes BTAM support for the remote 3270 display system or the local 3270 display system or both.

For information about other operands and other system generation macro instructions and about the system generation process, see IBM System/360 Operating System: System Generation, GC28-6554.

STORAGE ESTIMATES

FIXED MAIN STORAGE REQUIREMENTS

BTAM support for the local 3270 display system adds to the fixed main storage requirements for control program options described in IBM System/360 Operating System: Storage Estimates, GC28-6551; the nucleus of an MFT or MVT system is increased by 850 bytes for a new SVC and a new attention routine.

DYNAMIC MAIN STORAGE REQUIREMENTS

The dynamic main storage requirement is estimated by adding together a coding space estimate, a control information space estimate, a control block space estimate by line group, a control block space estimate by line, a control block space estimate by READ or WRITE macro instruction, and a channel program space estimate by line. Estimates for the 3270 display system are given in Figures 3270-10, 3270-11, 3270-12, 3270-13, 3270-14, and 3270-15.

For more information, see OS Storage Estimates, GC28-6551.

Description	Remote 3270 (in bytes)	Local 3270 (in bytes)
Primary requirement:		
• without buffer management	8,500	4,000
• with buffer pool support	8,950	4,450
• with dynamic buffering	10,164	NA[1]
Optional requirement:		
• online test	2,690	660
• line error print (LERPRT)	374	NA[1]
• line open (LOPEN)	530	530
• translate (TRNSLATE)	158	NA[1]
• change entry (CHGNTRY)[2]	352	72
• reset (RESETPL)[2]	600 or 256[3]	600 or 200[4]

[1] NA indicates not applicable.

[2] The estimate is for each macro instruction expansion.

[3] The lower estimate applies when POLLING is specified.

[4] The lower estimate applies when ATTENT is specified.

Figure 3270-10. Coding Space Estimates (3270 Display System)

Device Type	Bytes Required
Remote 3270 device	328
Local 3270 device	120

Figure 3270-11. Control Information Space Estimates for Each Device Type (3270 Display System)

Group Type	Control Blocks	Bytes Required
Remote 3270 line group	DCB, DEB	188 to 204 + 4 per line,
Local 3270 device group	DCB, IRB, IQE, DEB	260 + 4 per line

Figure 3270-12. Control Block Space for Each Line Group or Device Group (3270 Display System)

Line or Device Type	Control Block	Bytes Required
BSC Line for remote 3270 devices	IOB	64
	UCB	20
	Line error block	20
Local 3270 device	IOB	64
	UCB	40

Figure 3270-13. Control Block Space for each Line or Device (3270 Display System)

Macro Instruction	Control Block	Bytes Required
READ or WRITE for remote 3270	DECB	48
READ or WRITE for local 3270	DECB	40

Figure 3270-14. Control Block Space for Each READ or WRITE macro Instruction (3270 Display System)

Line or Device Type	Bytes Required
BSC Line for Remote 3270 Devices	88
Local 3270 device	24

Figure 3270-15. Channel Program Space Estimates for Each Line or Device (3270 Display System)

Library	Number of Directory Records	Number of Tracks Required								
		2301	2302	2303	2311	2314	2321	2305-1	2305-2	3330
SYS1.SVCLIB	18	7	NA	24	33	16	NA	14	10	10
SYS1.MACLIB (blocked)	2	13	65	66	70	37	120	20	18	22
SYS1.MACLIB (unblocked)	2	29	79	103	109	69	229	88	49	44
SYS1.TELCMLIB	1	2	2	2	3	2	4	2	2	2

Figure 3270-16. Auxiliary Storage Requirements for BTAM

AUXILIARY STORAGE REQUIREMENTS

BTAM support for the 3270 display system increases the auxiliary storage requirements described in _Storage Estimates_, GC28-6551, by adding to the SVC library, the macro library, and the subroutine libraries. Auxiliary storage requirements for BTAM with 3270 display support are given in Figure 3270-16.

STORAGE ESTIMATES FOR BTAM MODULES

Listed below are storage estimates for modified and new modules for BTAM support of the 3270 display system. See _Storage Estimates_, GC28-6551, for information about other BTAM modules.

Modified BTAM Modules

Routine	Module	Size (in Bytes)
BTAM Open Executor	IGG0193M	1024
BTAM Close Executor	IGG0203M	1024
Read/Write Channel Program Generator	IGG019MA	3158
Channel End/Abnormal End Appendage	IGG019MB	5516
Line Open	IECTLOPN	564
Online Test Control	IGG019MR	2560
Online Test Control (Start-Stop)	IGC006F	1024
Online Test Control (BSC)	IGC0D06F	1024
BTAM BSC ERP Control	IGE0004C	1024

New BTAM Modules

Routine	Module	Size (in Bytes)
BTAM Open Executor (for Local 3270)	IGG0194P	1024
BTAM Open Executor (for Local 3270)	IGG0194Q	1024
Channel End/Abnormal End Appendage (for Local 3270)	IGG019PA	455
BTAM Local 3270 SVC	IECTSVC	539
First-Level Attention Routine (for Local 3270)	IECTATEN	303
Second-Level Attention Routine (for Local 3270)	IGG019PG	160
Local Online Test Control	IGG019PI	480
EBCDIC Test Module (for 3270)	IGC0E06F	1024
EBCDIC Test Module (for 3270)	IGC0F06F	1024
ASCII Test Module (for Remote 3270)	IGC1006F	1024
ASCII Test Module (for Remote 3270)	IGC1106F	1024
Online Test Control (for Local 3270)	IGC1206F	1024
EBCDIC Test Module (for 3270)	IGC1306F	1024
EBCDIC Test Module (for 3270)	IGC1406F	1024
Error Post (for Remote 3270)	IGE0704B	1024
Local 3270 I/O Module	IGG019PH	143
3270 Scan Routine	IECTSCAN	4096

ONLINE TESTING

Test programs for the 3270 display system can be run under the System/360 Operating System using the Online Test Executive Program (OLTEP). For local 3270 control units and devices, an online test program and a BTAM application program can be executed concurrently; only the control unit or device being tested is unavailable to the application program. For remote 3270 control units and devices, a BTAM application program must end use of the line before an online test program can be executed.

For more information about OLTEP, see _IBM System/360 Operating System: Online Test Executive Program_, GC28-6650.

CONVERSION

For information about converting from 2260 display stations to the 3270 display system, see _IBM 2260 BTAM and 2260 GAM to IBM 3270 BTAM Conversion Guide_, GC27-6975.

BTAM provides facilities called error recovery procedures (ERP) that diagnose a variety of error conditions that occur during message transmission, and attempt to recover from those conditions that are considered recoverable so that transmission can continue. In addition to the diagnostic and recovery capabilities, error recovery procedures:

- Provide to the user program information about errors from which ERP could not recover, so that the program can act accordingly.

- Accumulate in special data sets, called the Statistical Data Recorder (SDR) and the Outboard Recorder (OBR), counts of certain kinds of errors; these are useful to the Customer Engineer in analyzing line and equipment troubles.

- Notify the operator at the central computer console (or some other console, if the system has the Multiple Console Support facility) of certain kinds of errors.

In addition to these ERP-provided functions, BTAM provides the capability of accumulating in a set of counters, collectively called a line error recording block (LERB), running totals of certain kinds of error conditions, and printing these totals at the central computer console (or some other console, if the system has the Multiple Console Support facility). These capabilities are provided by the LERB and LERPRT macro instructions.

Error recovery procedures handle errors arising from conditions at remote stations, on communications lines, and at transmission control units. The action with which ERP responds to an error condition depends on the kind of error, the type of I/O command (Poll, Read, Write, etc.) being executed when the error occurred, and the type of remote station involved. ERP analyzes the error condition, considering each of these factors, and performs the appropriate action. Error conditions are considered to be in one of three categories: irrecoverable, temporary, and permanent. An irrecoverable error is one that is inherently incapable of being corrected by program action; that is, some form of human intervention is required[1]. ERP does not attempt to recover such errors. When an irrecoverable error occurs, ERP sets indicators denoting the nature of the error in the data event control block (DECB) for the line involved in the I/O operation, notifies the operator at the central computer console by message IEA000I (see Appendix C for the format of this message), and records the occurence of the error in the Outboard Recorder (OBR), unless the error is a timeout, data check, or intervention required error. These three kinds of errors can optionally be recorded in the line error recording block.

If an error is not inherently irrecoverable, ERP attempts to recover from it, usually by reissuing the I/O command for which the error occurred or issuing other appropriate I/O commands to clear the condition. If ERP succeeds in clearing the condition, the error is said to be temporary. BTAM records its occurrence in the Statistical Data Recorder (SDR), and the user program continues normally. If ERP is unsuccessful in clearing the condition, the error is said to be permanent. BTAM records its occurrence in the Outboard Recorder (OBR), sets bits indicating the nature of the error in the DECB for the line involved in the I/O operation, and notifies the operator at the central computer console, by means of message IEA000I.

BTAM error recovery procedures attempt to clear error conditions up to two times, for operations on start-stop lines, and up to six times on BSC lines. Thus, permanent error conditions are indicated in the DECB after the I/O operation has been attempted three times (start-stop) or seven times (BSC). Irrecoverable errors are indicated in the DECB after only one attempt, since such errors are not retried.

Once an error condition has occurred and the ERP facility is trying to recover from it, any subsequent errors that may occur during the retries are not indicated in the DECB to avoid obscuring the original error

----

[1]An exception to this is the buffer-unavailable condition, which may occur when dynamic buffering is in use and the buffer pool becomes depleted. This condition (indicated by bit 4 in DECFLAGS), though not recoverable by ERP, can be anticipated by the user program, which can request retransmission of the lost message.

condition. Neither are subsequent errors recorded in the SDR, OBR, or LERB, nor are they indicated to the console operator. This ensures that only errors occurring during I/O operations issued by the user program are recorded.

Error recovery procedures are divided into basic functions and additional functions.

BASIC FUNCTIONS

- When an error occurs during an I/O operation that does not involve transmission of message text (these are called non-text errors), ERP retries the operation.

- ERP records each occurrence of a temporary error in the Statistical Data Recorder, and each occurrence of a permanent or irrecoverable error in the Outboard Recorder.

- ERP provides, in the data event control block for the line involved, information on (1) permanent errors, (2) irrecoverable errors, and (3) errors occurring during transmission of message text (called Read Text and Write Text errors) for which ERP does not attempt recovery. Recovery of Read Text and Write Text errors is an additional ERP function, described below.

- ERP sends a message to the operator at the console of the central computer (or some other console, if the system has the Multiple Console Support facility) for each permanent and irrecoverable error.

Basic ERP functions are optional for line groups involving start-stop terminals, but are required for line groups involving BSC stations. To obtain these basic functions for start-stop terminals, code EROPT=E in the DCB macro for the line group, or omit the EROPT operand. To omit the basic functions, code EROPT=N (valid only for start-stop line groups).

Coding EROPT=N prevents certain ERP routines from being included in the system, with a consequent saving in storage space. It is recommended that EROPT=N be coded for line groups for AT&T 83B3 and WU 115A terminals, for the reasons given in the explanation of the EROPT operand in Table 1.

ADDITIONAL FUNCTIONS

- When an error occurs during execution of a Read Text command (called a Read Text error), ERP will optionally retry the operation unless dynamic buffering is in use. This function is optional for the IBM 1050, 2740 (with the Record Checking feature), and 2260; it is unavailable for other start-stop terminals; and it is always required for BSC stations.

- When an error occurs during execution of a Write Text command (a Write Text error), ERP will optionally retry the operation unless dynamic buffering is in use. This function is optional for start-stop terminals, and results in an additional copy of the message text for each retry (except for messages sent to a card punch or tape punch of an IBM 1050 with the Line Correction feature, or to a 2260 with the Line Address feature); it is always required for BSC stations.

- ERP will optionally record, in the line error recording block, each occurrence of a data check, time-out, or intervention required error.

These three additional functions are specified as follows. To provide recovery attempts for Read Text errors (start-stop terminals), code EROPT=R in the DCB macro for the line group involved; to provide for recovery from Write Text errors (start-stop terminals), code EROPT=W. These EROPT parameters are ignored if dynamic buffering is specified for the line group, or if they are specified for a type of terminal for which recovery is not available. (It is not necessary to explicitly specify recovery attempts for Read Text and Write Text errors for BSC stations; the function is always performed for BSC.)

To provide for recording of errors in the line error recording block, code EROPT=C in the DCB macro, code a LERB macro to define the LERB, and code the name of the LERB macro in the LERB operand of the DCB.

Note: Errors occurring during transmission of test messages by the on-line test facility are not recorded in the LERB.

LERB (Line Error Recording Block) Macro Instruction

LERB defines for each line in a line group an area of main storage called a line error recording block. This block consists of a group of counters in which are kept cumulative totals of data check, intervention required, and non-text time-out errors, and of the number of transmissions. There are two sets of these counters for each communications line. The contents of one set, called the threshold counters, are incremented each time a transmission or an error occurs, until one of the counters reaches its threshold value, which is determined by the LERB macro. When the threshold value is reached, the contents of all four threshold counters are printed at the console of the central computer (message IEC801I; see Appendix C for the format). The contents of the threshold counters are added to the other set of counters, called cumulative counters or accumulators, and the threshold counters are reset to zero.

The contents of the accumulators are printed, and both sets of counters are optionally reset to zero when you issue a LERPRT (Line Error Recording Print) macro instruction.

Name	Operation	Operand
symbol	LERB	nlines[,{([transmct] [,datack],[,intreq] [,nontto])}...]

symbol
: is the name of the first line error recording block defined by this macro. It must be specified; it must also be coded in the LERB operand of the DCB macro for the line group.

nlines
: specifies the number of lines composing the line group for which the LERB macro is defining these blocks. The allowable range of values is 1 to 255.

transmct
: specifies the transmission count threshold: the number of consecutive transmissions that when reached causes the contents of the threshold counters to be printed, and the counters reset to zero. You may specify from 1 to 255; if you omit the operand, 255 is assumed.

datack
: specifies the data check error count threshold: the number of data checks that when reached causes the contents of the threshold counters to be printed, and the counters reset to zero. You may specify from 1 to the value of transmct. If you omit the operand, either 10 or the value of transmct, whichever is lower, is assumed.

intreq
: specifies the intervention required error count threshold: the number of intervention required errors that when reached causes the contents of the threshold counters to be printed, and the counters reset to zero. You may specify from 1 to the value of transmct. If you omit the operand, either 5 or the value of transmct, whichever is lower, is assumed.

nontto
: specifies the non-text time-out error count threshold: the number of non-text time-out errors that when reached causes the contents of the threshold counters to be printed and the counters reset to zero. You may specify from 1 to the value of transmct. If you omit the operand, either 5 or the value of transmct, whichever is lower, is assumed.

Of these operands, only <u>nlines</u> is mandatory. If you code only this operand, the default threshold counts of 255, 10, 5 and 5 are assumed for each of the lines in the line group. If you code threshold counts for one line, the same counts are assumed for all successive lines for which you do not code threshold counts; the counts are effective until you code another set of threshold counts.

<u>Example</u>:

```
LG1LERB   LERB 10,,,(200,20,,7),,,
               (240,20,25,10)
```

This LERB macro provides line error recording blocks for ten lines:

- The first two lines have assumed threshold values of 255 (transmct), 10 (datack), 5 (intreq), and 5 (nontto).
- The next three lines have explicit transmct, datack, and nontto threshold values of 200, 20, and 7, and an assumed intreq threshold value of 5.
- The remaining five lines have explicit threshold values of 240, 20, 25, and 10.

  Omit the parentheses if you omit all the suboperands (transmct, etc.) for a line, as has been done for lines 1, 2, 4, 5, 7, 8, 9 and 10 in the example. Also omit trailing commas (as for lines 7, 8, 9, and 10 above).

LERPRT (Line Error Recording Print) Macro Instruction

LERPRT causes the current contents of the cumulative counters (accumulators) for a line to be printed at the console of the central computer or, in systems with Multiple Console Support, at some other user-designated console. (The message number is IEC802I; see Appendix C for the format.) As explained under the LERB macro, the contents of the four threshold counters for the line are added to the accumulator each time one of the threshold counters reaches its threshold value. The contents of the threshold counters are also added to the accumulators (and the threshold counters are reset) when you issue the LERPRT macro. You may also specify in LERPRT that the threshold counters and accumulators for the line be reset to zero.

Caution: Issue a LERPRT macro for a line only if the line group containing that line is open.

Name	Operation	Operand
[symbol]	LERPRT	dcbaddr[,rln][,cid] [,CLEAR=YES / ,CLEAR=NO]

dcbaddr
: specifies the address of the DCB for the line group containing the line for which the contents of the accumulators are to be printed.

rln
: specifies the relative line number of the line involved. rln cannot exceed the number of lines contained in the line group (as indicated in the nlines operand of the LERB macro). If you omit this operand, all non-zero values of the accumulators for all lines in the group are printed.

cid (applicable only to systems having the Multiple Console Support facility)

specifies the address of a byte containing the identification of the console at which the message is to be printed. If you omit this operand when the system includes the MCS facility, the message is printed at consoles having a routing code of 8. If you specify an invalid console ID, the message is printed at consoles having a routing code of 1.

CLEAR=YES
: specifies that the threshold counters and accumulators be reset to zero after the contents of the accumulators are printed. If you omit this operand, CLEAR=YES is assumed.

CLEAR=NO
: specifies that the threshold counters and accumulators are not to be reset after the contents of the accumulators are printed.

ERROR DETECTION AND ANALYSIS

To determine what action to take when an error occurs requires that the error condition be analyzed. This is true whether error recovery is to be performed by ERP routines or by the user program. If BTAM ERP is used, all error analysis and recovery attempts are performed before the Read or Write operation is posted as complete in the event control block (ECB) for the line. All user program analysis and recovery attempts, whether in addition to BTAM ERP actions or in place of them, occur after the Read or Write operation is posted as complete. Thus the user program analysis and recovery routine should receive control following each completion of a Read or Write operation.

As mentioned previously, BTAM sets various error indicators in the DECB. Some of these are set before ERP routines gain control, others are set by the ERP routines. The DECB fields containing error indicators are:

- DECSDECB. This is the address of the event control block, which is the first fullword of the DECB. The first byte contains the completion code for the operation.
- DECFLAGS. A one-byte field containing flags that BTAM sets at the conclusion of a Read or Write operation. Some of these flags represent not errors but exceptional conditions of interest to the user program. These flags are set regardless of whether the operation was completed successfully.
- DECERRST. A one-byte field containing flags indicating the kind of I/O error that occurred. These flags are set only in the event of a true I/O error (as opposed to an exceptional condition) as indicated in the event control block by a completion code of X'41'.
- DECCSWST. A halfword field containing the status indicators set in the channel status word (CSW) at the conclusion of the Read or Write operation.

* DECSENS0. A one-byte field containing the sense information returned by the transmission control unit at the conclusion of the I/O operation that resulted in a Unit Check error.

User Program Analysis Procedure

A recommended procedure for checking the results of a Read or Write operation is as follows.

First determine whether the operation was completed successfully or unsuccessfully. A completion code of X'7F' in the event control block (DECSDECB) indicates successful completion, and a code of X'41' indicates unsuccessful completion. A third completion code, X'48', indicates that the Read or Write operation was halted as a result of issuing a RESETPL or CLOSE macro while the operation was in progress.

Normal Completion

If the completion code is X'7F', check the DECFLAGS field. If it contains all zeros, no exceptional condition has occurred, and the program can execute whatever Read or Write operation would normally follow the completed operation. If DECFLAGS does not contain all zeros, check each of the appropriate flags to determine the exceptional condition, then take suitable action.

Abnormal Completion

If the completion code is X'41', determine the nature of the I/O error as follows. Check the DECFLAGS and DECERRST fields. If DECERRST does not contain all zeros, check the appropriate bits from among bits 0-4. (Bit 2 is applicable unless BTAM ERP routines are not present in the system; bit 3 is applicable only if the TCU is an IBM 2701.)

If both DECFLAGS and DECERRST contain all zeros, check the bits in the CSW status field, DECCSWST, in the order indicated in the priority column of Table 13. If the Unit Check bit is on, check the sense bits in DECSENS0 in the order indicated in Table 14. Then take appropriate action. These last two fields, DECCSWST and DECSENS0, contain the same information as was checked by the ERP routines, if the system includes the ERP facility.

The preceding discussion mentioned checking the "appropriate" bits; this simply means those bits that could possibly be set in a given situation. For example, it would be appropriate to check bit 0 of DECFLAGS only if you were analyzing the result of an operation on a BSC line, as bit 0 is not used in start-stop operations.

Many factors are involved in determining what action is suitable in response to a given kind of error or exceptional condition.

For exceptional conditions occurring in conjunction with a successful completion of a Read or Write operation, the question is not of retrying an operation that failed, but rather of determining what kind of Read or Write operation should be executed next.

Some of the factors to be considered in determining what program action to take in the event of an error or exceptional condition are the type of Read or Write operation (e.g. Read Initial, Write Continue) and the specific command (e.g. Poll, Read Text, Write Response) being executed when the error occurred, and the type of remote station or line involved. The characteristics of the application will determine other factors to be considered. In any event, it is inadvisable to pursue the same procedure that the BTAM ERP facility does, for any given error condition. The tables and explanations are intended to show what BTAM ERP has already done to recover, or to guide you in writing your own recovery procedures; they are not intended to suggest user analysis actions once error recovery has been attempted. The next chapter suggests, for BSC lines, appropriate macros to issue after BTAM posts exceptional or error conditions.

In general, BTAM error recovery procedures anticipate all of the possible conditions from which recovery may be possible, and in each case takes appropriate action to achieve recovery. For this reason, use of the BTAM-provided error recovery procedures is highly recommended, and saves much coding effort for the application programmer. In many applications, the appropriate action for the user program to take when ERP fails to recover is simply to ignore the error condition and resend the same message text with the next Read or Write operation, or to defer further operations on the line until an equipment failure or abnormal condition can be corrected. Usually, it is appropriate for the program to notify the operator at the central computer console or teleprocessing console of the condition.

The remainder of this chapter provides detailed explanations of what actions BTAM error recovery procedures perform for each of the error and exceptional conditions, and for each of the commands and types of remote stations for which the error can occur. This information is provided for users wishing to write their own error analysis and recovery routines, for use either in place of or in addition to the BTAM-provided ERP facility. As mentioned previously, an error analysis routine usually must consider the type of Read or Write operation and the type of channel command on which the error or exceptional condition occurred. Three fields in the DECB contain this information:

- DECTYPE indicates the kind of Read or Write operation being executed.

  Note: Certain error conditions cause BTAM ERP to modify the DECTYPE field, so that this field has a different value at the end of the Read or Write operation from the value it initially had. No other user-specified DECB field is modified by ERP.

- DECCMCOD indicates the specific type of channel program command on which the error occurred.

- DECTPCOD indicates the TP Operation Code associated with the command. Each BTAM channel command contains this code, in byte five of the channel command word (CCW). This code, which is not present in CCWs for other (non-TP) environments, has no effect on channel operations. It is in effect an extension of the command code, and identifies the purpose of the command.

The meanings of the bits in each of these fields are given in Appendix B. In these descriptions, and in the descriptions of the BTAM ERP actions below, references are made to types of I/O operations, equipment conditions, and indicators that are not defined elsewhere in this publication, as they relate to the operational details of specific types of transmission control units and related equipment. Understanding of these references and writing of an error recovery routine requires a knowledge of the functional complexities of this equipment, which is beyond the scope of this publication to impart. Information on transmission control units may be found in the publications listed under that heading, at the front of this publication.

Table 13. User Program Status Analysis

Priority	CSW Bit	Condition	Remarks
1	45	Channel Control Check	
2	46	Interface Control Check	
2	44	Channel Data Check	
3	32	Attention	
3	33	Status Modifier	
3	34	Control Unit End	
3	35	Busy	
4	38	Unit Check	Check Sense Bits
5	47	Chaining Check	
6	42	Program Check	Program Error
6	43	Protection Check	Program Error
7	39	Unit Exception	Possible Error
8	41	Incorrect Length	Possible Error

Table 14. User Program Sense Byte Analysis

Priority	Sense Bit	Condition
1	3	Equipment Check
2	6	Lost Data
3	7	Timeout
4	1	Intervention Required
5	2	Bus Out
6	4	Data Check
7	5	Overrun
8	0	Command Reject

BTAM ERP ERROR ANALYSIS AND RECOVERY ACTIONS

BTAM routines analyze error and exceptional conditions by examining the status field of the channel status word (CSW) and in the case of a Unit Check error, by examining the sense information provided by the TCU at the end of the I/O operation.

More than one indication may be set in the CSW when an error condition occurs. Generally, only one of these indicators properly describes the condition; other indicators reflect secondary effects. Similarly, transmission control unit errors can cause more than one sense bit to be set. To ensure that the primary condition is recognized and acted upon, priority schemes determine the order in which status and sense bits are tested.

Tables and explanations of actions are shown separately for start-stop and BSC operations, as the actions differ considerably for these two categories.

"Should-Not-Occur" Errors

Among the error conditions causing BTAM to pass control to ERP are those which are undefined for a particular command or combination of hardware conditions. An overrun error, for example, has no meaning for a Write command.

In some instances, however, ERP can recover even from undefined errors. Consider, for example, a lost data error occurring on a Dial command issued for a start-stop line. Although this condition, like lost data on a Write command, is not defined (and is therefore not mentioned in the SRL publications for the IBM 2701, 2702, and 2703), ERP does try to recover by reexecuting the Dial command (see Table 19 and ERP action 20.) After two unsuccessful retries, ERP disables the line, notifies the operator, and records the error in the Outboard Recorder.

Note that ERP does not attempt recovery when a lost data error occurs on a Write command; rather, ERP immediately indicates a "should-not-occur" error has occurred. In Table 19, Write falls in the "all other commands" category; see action 11.

Table 15 shows for various commands the error conditions OS BTAM ERP considers to be "should-not-occur" errors and therefore does not attempt recovery.

Table 15. Should-Not-Occur Error Conditions Posted by BTAM ERP

Error Condition	Write	Read	Inhibit	Prepare	Sense	Disable	Enable	Dial	Poll
Lost Data	•			•[1]	•	•	•		•
Timeout	•[2]		•		•				
Intervention Required					•	•	•		
Bus Out Check		•	•	•	•	•	•		
Data Check	•[3]			•	•	•	•	•	•
Overrun	•			•	•	•	•	•	•[4]
Unit Exception				•[6]	•	•	•[6]	•[5]	

[1]Should-not-occur error only for start-stop lines.
[2]Not a should-not-occur error for any Write command that sends transparent text (BSC lines), or for a 2260/2848 (start-stop lines).
[3]Should-not-occur error only for Write operations on BSC lines.
[4]Should-not-occur error only if TCU is a 2701, for start-stop lines; always a should-not-occur error for BSC lines.
[5]Should-not-occur error only if TCU is a 2701.
[6]Not a should-not-occur error if TCU is a 2703 and user program issued RESETPL macro.

START-STOP ERROR RECOVERY PROCEDURES

In the tables, figures in parentheses following the name of a command represent the TP-OP code of the command, as appearing in the DECTPCOD field of the DECB at completion of a Read or Write operation.

Table 16. Status Analysis -- Start-Stop

CSW Bit	Condition	Remarks	Priority	Action
45	Channel Control Check		1	1
46	Interface Control Check		2	1
44	Channel Data Check		2	1
32	Attention		3	2
33	Status Modifier		3	2
34	Control Unit End		3	2
35	Busy		3	2
38	Unit Check	Check Sense Bits	4	(See Table 17)
47	Chaining Check		5	3
42	Program Check	Program Error	6	4
43	Protection Check	Program Error	6	4
39	Unit Exception	Possible Error	7	(See Table 18)
41	Incorrect Length	Possible Error	8	11

Table 17. Sense Byte Analysis --Start-Stop

Priority	Sense Bit	Condition	Action
1	3	Equipment Check	5
2	6	Lost Data	See Table 19
3	7	Timeout	See Table 20
4	1	Intervention Required	See Table 21
5	2	Bus Out	See Table 22
6	4	Data Check	See Table 23
7	5	Overrun	See Table 24
8	0	Command Reject	See Table 25

Table 18. Unit Exception -- Start-Stop

Command	Action
Write or Poll (03)	
2741	7
83B3 or 115A (Telegraph adapter type I)	8
all other terminals	9
Read	
Read response to addressing (06)	
Terminal adapter type I	
2740	10
all other terminals	11 (SNO)
All other types of adapters	11 (SNO)
Read response to text (20)	
2740 with station control & checking, or 2260	
Write text retries specified (EROPT=W)	12
Write text retries not specified	13
all other terminals	11 (SNO)
All other Reads	11 (SNO)
All other commands	11 (SNO)

Table 19. Lost Data -- Start-Stop

Command	Action
Read	
Read ID response (07)	14
Read response to addressing (06)	15
Read Text (11)	
If residual count = 0	
83B3 or 115A (Telegraph adapter type I)	16
all other terminals	17
If residual count not = 0	18 (SNO)
Read response to text (20)	19
Dial (01)	20
All other commands	11 (SNO)

Table 20. Timeout -- Start-Stop

Command	Action
Write	
2260 or 2265 (Terminal Adapter Type III)	21
all other terminals	22 (SNO)
Read (or Inhibit)	
Read response to addressing (06)	
83B3	23
all other terminals	24
Read ID response (07)	25
Read response to polling (05)	
First command is Disable	23
First command is not Disable	25
Read Index (0A)	
First command is Disable	23
First command is not Disable	26
Read text (11)	27
Read response to text (20)	
Write text retries specified (EROPT=W)	
Operation is Write Continue	13
Operation is not Write Continue	
First command is Disable	28
First command is not Disable	24
Write text retries not specified	13
Disable (01)	29
Enable (01)	
preceded by Disable	24
not preceded by Disable	30(SNO)
Prepare (01)	25
Dial (01)	31
Poll (03)	32
All other commands	33 (SNO)

Table 21. Intervention Required -- Start-stop

Command	Action
Write or Break (23)	34
Read (or Inhibit)	
Read Index (0A)	26
Read response to polling (0A)	
Operation is Write TTV	35
Operation is not Write TTV	36
Read response to text (20)	
Read text retries specified (EROPT=R)	
Operation is Write Continue	37
Operation is not Write Continue	24
Read text retries not specified	37
All other Reads	39
Dial (01)	40
Prepare (01)	
2741	61
All other terminals	36
Poll (03)	24
All other commands	41 (SNO)

Table 22. Bus Out Check -- Start-Stop

Command	Action
Write	
prior to Write text	46
Write text (11)	
Write text retries specified (EROPT=W)	
Failing command is last command or next command is not Read response to text	46
Next command is Read response to text	45
Write text retries not specified	44
following Write text	45
Dial (01)	46
Poll (03)	46
All other commands	11 (SNO)

Table 23. Data Check -- Start-Stop

Command	Action
Write	
83B3, 115A, or TWX (33,35) (Telegraph adapter type I or II)	
Error occurred during text transfer	
Write retries specified (EROPT=W)	24
Write retries not specified	39
Error did not occur during text transfer	24
World Trade Telegraph terminals (WTT Adapter)	37
Read (or Inhibit)	
Read ID response (07)	48
Read response to polling (0A) or any other Read command prior to text transfer	
First command in channel program is Disable	49
First command in channel program is not Disable	50
Read text (11)	
Dynamic buffering is used	51
Dynamic buffering is not used	
Read text retries specified (EROPT=R)	
Operation is Read Repeat	24
Operation is Write Continue Conversational	53
All other operations	54
Read text retries not specified	51
Read response to text (20)	
Operation is Write Continue	48
All other operations	52 (SNO)
Remote terminal is 2740 with 2760 feature, and operation is Write TIO, TCO, or TVO	55
Poll (03)	56
All other commands	52 (SNO)

Table 24. Overrun -- Start-Stop

Command	Action
Read	
Read Index (0A)	57
Read response to text (20)	58
All other Reads	
Dynamic buffering is used	59
Dynamic buffering is not used	
Read text retries are specified (EROPT=R)	60
Read text retries are not specified	59
Poll (03)	
TCU is 2702	57
TCU is not 2702	11 (SNO)
Inhibit (11)	
Read text retries are specified (EROPT=R)	60
Read text retries are not specified	59
All other commands	11 (SNO)

Table 25. Command Reject -- Start-Stop

Command	Action
Poll (03)	11 (SNO)
All other commands	9

ERROR RECOVERY ACTIONS FOR START-STOP OPERATIONS

In the descriptions of the actions performed by BTAM error recovery procedures, the phrase "if applicable", applied to the action of recording occurrences in the line error recording block (LERB), means that the error is so recorded only if it is a data check, intervention required, or non-text time-out error, and if you have specified that errors be recorded by appropriately coding the EROPT and LERB operands of the DCB macro instruction.

The phrase "ERP notifies the console operator" means that ERP writes message IEA000I to the console of the central computer, and/or to some other console (e.g., teleprocessing console) if the Multiple Console Support facility is in use. This allows the operator to take whatever action is necessary to correct the condition. See Appendix C for the format of this message.

The phrase "ERP posts the operation as complete-with-error" means that ERP sets the post flag in the ECB for the line to indicate conclusion of the operation, and sets a completion code of X'41' in the ECB, to indicate that an I/O error occurred.

1 If the Channel Check Handler (CCH) of the Recovery Management Support facility has been included in the operating system during system generation, ERP forces a permanent error condition by setting the Unit Check and Equipment Check sense bits in the sense byte. This causes ERP for this condition to notify the console operator. Operations can proceed normally for line groups using other channels. If the CCH facility is not included in the operating system, the condition is recorded in the System Environment Recorder (SER) and the Supervisor enters Wait state, as the error is too serious to allow further operations.

2 ERP sets the should-not-occur bit (bit 1) in DECERRST, posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.

3 If the error occurred on a Read command, ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR. If the error did not occur on a Read command, ERP sets the should-not-occur bit (bit 1) in DECERRST, posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.

4 ERP notifies the console operator. The error is probably a program error.

5 ERP notifies the console operator. The error is caused by control unit failure, and the communications line involved should be considered inoperative.

6 ---

7 ERP restarts the channel program at the Write or Poll command. On the third occurrence of the error, ERP posts the operation complete-with-error and records the occurrence in the OBR.

8 ERP executes a Break command, then restarts the channel program. On the third occurrence of the error, ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.

The error indicates that data is being received from the line without a command.

9 ERP executes a Read Skip command. If the Read Skip is successful, ERP restarts the channel program at the failing command. On the third occurrence of the error, ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.

The error indicates that data is being received from the line without a command.

10 The Unit Exception condition is normal for a 2740 and indicates receipt of a positive or negative response (Y or N). If Y was received, ERP turns off the Unit Exception bit and restarts the channel program at the next command. If N was received, ERP posts completion with or without error. Receipt of a character other than Y or N is a should-not-occur condition; ERP sets the should-not-occur bit (bit 1) in DECERRST and posts the operation complete-with-error.

11 ERP sets the should-not-occur bit (bit 1) in DECERRST, posts the operation complete-with-error, notifies

the console operator, and records the occurrence in the OBR.

12 ERP executes a Write Continue channel program to resend the same message text. On the third occurrence of this error, ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.

The error indicates a buffer overflow.

13 ERP posts the operation complete-with-error.

14 ERP restarts the channel program at the failing command. On the third occurrence of this error, ERP disables the line, posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.

15 ERP restarts the channel program. On the third occurrence of the error, ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.

16 ERP executes a Break command, posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.

The error indicates that the input message was larger than the input area specified in the READ macro.

17 ERP executes a Read Skip command, posts the operation complete- with-error, notifies the console operator, and records the occurrence in the OBR.

The error indicates that the input message was larger than the input area specified in the READ macro.

18 ERP sets the should-not-occur bit (bit 1) in DECERRST and posts the operation complete-with-error.

19 ERP executes a Read Skip command, posts the operation complete- with-error, and notifies the console operator, and records the occurrence in the OBR.

20 ERP restarts the channel program at the Dial command. On the third occurrence of the error, ERP disables the line, posts the operation complete-with-error, notifies the console operator, and records the error in the OBR.

The error indicates that the Dial command was sent to a line that was already in the "off-hook" condition.

21 ERP builds and executes a Reset channel program. The error occurrence is recorded in the LERB (if applicable).

22 ERP sets the should-not-occur bit (bit 1) in DECERRST, posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR. The occurrence is recorded in the LERB (if applicable).

23 ERP restarts the channel program at the third command. On the third occurrence of the error, ERP posts the operation complete- with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable).

24 ERP restarts the channel program. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable).

25 ERP restarts the channel program at the failing command. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable).

26 ERP restarts the channel program to resend the polling sequence. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable).

27 ERP posts the operation complete-with-error. The error indicates that no text was received or that the elapsed time between successive text characters exceeded about 28 seconds (the intercharacter timeout interval).

28 ERP restarts the channel program at the third command. On the third occurrence of the error, ERP posts the operation complete-with-error.

29 ERP restarts the channel program at the failing command. On the third occurrence of the error, ERP posts

the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable).

The error indicates that the data set (modem) is failing to disconnect.

30 ERP sets the should-not-occur bit (bit 1) in DECERRST, posts the operation complete-with-error, and notifies the console operator.

31 ERP restarts the channel program at the failing command. On the third occurrence of the error, ERP posts the operation complete- with-error and notifies the console operator. Each occurrence of the error is recorded in the LERB (if applicable).

The error indicates that the remote station is not answering, when dialed, in the time allotted.

32 ERP restarts the channel program to resend the polling sequence. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable).

The error indicates that no response was received from the remote station.

33 ERP sets the should-not-occur bit (bit 1) in DECERRST and posts the operation complete-with-error. The occurrence is recorded in the LERB (if applicable).

34 ERP sets the retry count to maximum and posts the operation complete-with-error. The occurrence is recorded in the LERB (if applicable).

The error indicates that the addressed line has not been enabled.

35 ERP restarts the channel program at the third command to resend the polling sequence. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable).

36 ERP restarts the channel program at the beginning. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable).

37 ERP posts the operation complete-with-error and notifies the console operator. The occurrence is recorded in the LERB (if applicable).

38 ---

39 ERP posts the operation complete-with-error and notifies the console operator.

40 ERP restarts the channel program at the failing command. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator.

The error indicates that the Automatic Calling Unit power is off or that the addressed line is not connected to an Auto Call adapter.

41 ERP sets the should-not-occur bit (bit 1) in DECERRST, posts the operation complete-with-error, and notifies the console operator. The occurrence is recorded in the LERB (if applicable).

42 ---

43 ---

44 ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.

The error is a parity error within either the command or the text data.

45 ERP records the error occurrence in the SDR and restarts the channel program at the failing command. On the third occurrence of the error, ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.

The error is a parity error within either the command or the text data.

46 ERP records the error occurrence in the SDR and restarts the channel program. On the third occurrence of the error, ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.

The error is a parity error within either the command or the text data.

47 ---

48 ERP restarts the channel program. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable). (See Note 1.)

49 ERP restarts the channel program at the third command. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable). (See Note 1.)

50 ERP restarts the channel program to resend the polling sequence. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable). (See Note 1.)

51 ERP posts the operation complete-with-error. The occurrence is recorded in the LERB (if applicable). (See Note 1.)

52 ERP sets the should-not-occur bit (bit 1) in DECERRST, posts the operation complete-with-error, and notifies the console operator. (See Note 1.)

53 ERP restarts the channel program at the Read Conversational part. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable). (See Note 1.)

54 ERP builds and executes a Read Repeat channel program. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable). (See Note 1.)

55 ERP restarts the channel program at the Write EOA PRE o command. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator. Each occurrence is recorded in the LERB (if applicable). (See Note 1.)

56 ERP restarts the channel program to resend the polling sequence. On the third occurrence of the error, ERP posts the operation complete-with-error. Each occurrence is recorded in the LERB (if applicable).

57 ERP restarts the channel program to resend the polling sequence. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator.

58 ERP restarts the channel program at the failing command. On the third occurrence of the error, ERP posts the operation complete-with-error and notifies the console operator.

59 ERP posts the operation complete-with-error, notifies the console operator, and records the occurrence in the OBR.

60 ERP builds and executes a Read Repeat channel program and records the error occurrence in the SDR. On the third occurrence of the error, ERP posts the operation complete- with-error, notifies the console operator, and records the occurrence in the OBR.

61 ERP issues Write EOT and restarts the Channel Program at the failing command.

Note 1: The error is one of the following:

- A VRC (parity) error was detected in one or more of the received characters.
- An LRC error was detected; i.e., the LRC character received from the remote station did not match the LRC value generated by the transmission control unit.
- A negative response was received as a response to text.
- The communication line was in the "space" condition at stop-bit time, indicating that the TCU was out of synchronism.

BSC ERROR RECOVERY PROCEDURES

The BTAM Channel End/Abnormal End Appendage will receive control from the supervisor following an I/O interrupt or after an ERP routine issues a SVC 15 with no retry specified (i.e., with bit 2 of IOBFLAG1 set to 0). When it receives control, the appendage makes an analysis of such things as the CSW information, the condition code, the operation in progress, the response received, etc., in order to determine the specific action to be performed based on the conditions existing.

The appendage passes control back to the supervisor as follows:

- At 0 + Register 14 - the channel program is posted complete and the request element is made available. This is the so-called "normal return."
- At 4 + Register 14 - the channel program is not posted complete but the request element is made available. This return is made for on-line test or when a SAD (Set Address) or Enable error occurs.
- At 8 + Register 14 - the channel program is not posted complete and its request element is placed back on the request queue so the program can be retried. This return is used when the channel program is to be restarted.

The supervisor then determines where control is to be passed next; if the IOB exception bit is on (bit 5 in IOBFLAG1=1) and the DCB indicates that basic error recovery procedures are provided (i.e., bit 7 in DCBERROP=0) for this line group, control is passed to the BSC ERP control routine. The control routine will also receive control from the supervisor following an I/O interrupt when an ERP channel program has been initiated and the ERP routines, rather than the Channel End/Abnormal End Appendage, are to analyze the results of the operation.

The BSC ERP control routine determines which ERP routine is to receive control, causes it to be loaded if necessary, and transfers control to it. Recovery actions, as indicated in the tables following, are then performed.

By using Table 26, you can locate the set of conditions for which you wish to determine the ERP action. In some cases, it will be necessary to consult more than one table to trace the complete sequence of the actions, as when ERP sets up a special return code to indicate the existence of a specific situation as it goes through a multi-step recovery procedure.

You should note that the tables present in summary form the actions performed by ERP; details such as incrementing the retry count and testing for a need for LERB recording are not shown.

In the tables, the values in parentheses following each command, e.g. Read Text (11), is the TP-Op code for that command, in hexadecimal.

Where the phrase "proceed with error posting" appears, see Table 40 to determine the actions taken by the Error Post routine.

Table 27 serves as an entry point for tracing the ERP-initiated recovery actions. In many cases, it refers to another table for further definition of recovery actions. Tables 39 and 40 indicate the result of control being passed to the Special Return routine and the Error Post routine, respectively.

Two examples of the use of the tables are as follows:

Example 1: The accumulated block check character (bcc) does not match the bcc received following the ETB or ETX ending a text block, on a Read Text command, causing a status indication of Channel End/Device End/Unit Check, with Data Check indicated in the sense byte.

Step 1 - Refer to the Unit Check section of Table 27. Since Channel End and Device End are on, you are referred to Table 31 to find further actions based on the results of the ERP analysis of the sense information.

Step 2 - Refer to Table 31, which refers you to Table 35.

Step 3 - Refer to Table 35, where the ERP actions for various commands are described. Since the error occurred on a Read Text command, the ERP action taken depends on whether or not dynamic buffering is being used. Assuming that it is not, ERP will build a channel program to write NAK and then transfer-in-channel (TIC) back to the Read Text command. ("Failing CCW" refers to the CCW on which the interrupt occurred.)

Example 2: A transmission causes an incoming ETB or ETX character to be distorted so that it is not recognized as a control character, the bcc characters are considered data characters and sent into main storage, exhausting the count in the Read Text CCW. The status indication is

Channel End/Device End/Unit Check, with Lost Data indicated in the sense byte.

Step 1 - Same as in example 1.

Step 2 - Refer to Table 31, which, for a Lost Data condition, refers you to Table 36.

Step 3 - Refer to Table 36, which indicates that, assuming dynamic buffering is not being used, a special return code (X'82') is set up in IOBWORK + 1 (one of the two locations in the Input/Output Block (IOB) where special codes are placed for later analysis by the Special Return routine). Then a channel program is generated to read the ENQ that the transmitting station will send when it does not receive a response to the block of text. When the channel program ends as a result of receiving the ENQ, the ERP Special Return routine is entered.

Step 4 - Refer to Table 39, Part B, which indicates that ERP builds a channel program to write NAK and then restarts the original channel program at the Read Text command.

Table 26. Index to BSC ERP Tables

Table	Description
27	Status Analysis
28	Channel Data Check
29	Equipment Check
30	Command Reject
31	Sense Byte Analysis
32	Bus Out
33	Overrun
34	Intervention Required
35	Data Check
36	Lost Data
37	Timeout
38	Unit Exception
39	Special Return Codes
40	Error Post Actions

Table 27. Status Analysis -- BSC

Status	Action
Attention Status Modifier Control Unit End Busy	1. Set "should Not Occur" bit in DECERRST. 2. Update statistics table. 3. See Table 29 for further actions.
Start I/O Condition Code = 1 (CSW Stored)	1. Update statistics table. 2. See Table 29 for further actions.
Channel Data Check	1. Update statistics table. 2. See Table 28 for further actions.
Program Check Protection Check Chaining Check	1. Set indicator to cause recording of occurrence in the Outboard Recorder (OBR). 2. Notify the console operator.
Unit Check	1. If Channel End and Device End status bits are both off, update the statistics table. See Table 29 for further actions. 2. If Channel End and Device End status bits are <u>not</u> both off, analyze the sense information. See Table 31 for actions resulting from this analysis.
Start I/O Condition Code = 3 (Not Operational)	1. Write operator message - "IEC0804A xxx CONTROL UNIT NOT OPERATIONAL, REPLY CONT OR POST" where xxx is the line address of the line involved. 2. If reply is "CONT" retry the failing channel program; if reply is "POST", post ECB complete with permanent I/O error; if reply is neither "CONT" nor "POST", repeat the message.
Unit Exception	This status bit can be turned on by equipment or by BTAM. See Table 38 for action taken when this status bit is on.

Table 28. Channel Data Check -- BSC

Command	Action
Write Text (11)	Indicate a permanent I/O error, then proceed with error posting.
Any Write except Write Text	If retry limit (7) has been reached, proceed with error posting; if not, retry the failing CCW.
Read Text (11) (Dynamic Buffering Not Used)	If retry limit (7) has been reached, proceed with error posting; if not, Write NAK and TIC to the failing CCW.
Read Text (11) (Dynamic Buffering Used)	Indicate a permanent I/O error, then proceed with error posting.
Read ENQ (0B)	If retry limit (7) has been reached, proceed with error posting; if not, retry the failing CCW.
Read Response to ENQ (0C)	Write ENQ and TIC to failing CCW.
Read Response to Text (25)	Write ENQ and TIC to failing CCW.
All other Reads	Set up special return code X'80' in IOBWORK+1 and generate a channel program to read a response.

Table 29. Equipment Check -- BSC

Command	Action
Write Text (11) (Dynamic Buffering not used)	Write ENQ and TIC to the CCW following the failing CCW (the Read Response to Text command).
Read Text (11) (Dynamic Buffering not used)	Set up special return code X'80' in IOBWORK+1 and generate a Read response channel program.
Read or Write Text (11) (Dynamic Buffering used)	Indicate permanent I/O error, then proceed with error posting.
Any command not during text transfer (i.e., any TP-Op code other than 11)	If retry limit has been reached, indicate a permanent I/O error and proceed with error posting; if not, restart the channel program.

Table 30. Command Reject -- BSC

Command	Action
Read Response to Text (25)	If retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, set up special return code X'04' in IOBERRCT+1 and write DLE ENQ.
Any command with a Special Return code of X'04'	Check for Channel End and Device End status only. If both bits are on, clear the special return indicator and return to the supervisor; if not, indicate permanent I/O error, then proceed with error posting.
All other commands	Indicate permanent I/O error, then proceed with error posting.

Table 31. Sense Byte Analysis -- BSC

Sense Bit	Condition	Action
0	Command Reject	See Table 30
1	Intervention Required	See Table 34
2	Bus Out Check	See Table 32
3	Equipment Check	See Table 29
4	Data Check	See Table 35
5	Overrun	See Table 33
6	Lost Data	See Table 36
7	Timeout	See Table 37

Table 32. Bus Out -- BSC

Command	Action
Dial (01)	If retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, disable the line and TIC to the first CCW.
Poll (03)	If retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, restart the channel program at the first CCW and start polling with the failing station.
Write -- Prior to Text Transfer (TP-Op Code less than 10)	Check to see if this is a Write EOT command (TP-Op Code 02). If so, restart the channel program at the failing CCW; if not, restart the channel program at the CCW following the failing CCW.
Write -- During Text Transfer (11)	Check to see if the residual count is equal to the original count. If so, restart the channel program at the failing CCW; if not indicate a permanent I/O error, then proceed with error posting.
All other Writes	Check to see if this is the last CCW in the channel program. If so, restart the channel program at the failing CCW; if not, restart the channel program at the CCW following the failing CCW.
Any Read Command	Check to see if the residual count is equal to the original count. If so, restart the channel program at the failing CCW; if not, indicate a permanent I/O error then proceed with error posting.
All other commands	Restart the channel program at the failing CCW.

Table 33. Overrun -- BSC

Command	Action
Read ENQ (08)	If retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, restart the channel program at the failing CCW.
Read Response to ENQ (0C)	If retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, restart the channel program at the CCW preceding the failing CCW.
Read Response to Text (25)	If retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, generate a channel program to Write ENQ, then TIC to the failing CCW.
Read Text (11)	If dynamic buffering is specified, indicate a permanent I/O error, then proceed with error posting; if not (and if retry attempts have not been exhausted), generate a channel program to Write NAK, then TIC to the failing CCW.
All other commands	Set "should not occur" bit in DECERRST, indicate a permanent I/O error, then proceed with error posting.

Table 34. Intervention Required -- BSC

Command	Action
	Each of the actions described below is preceded by a test to see whether the retry limit (7) has been reached. If so, a permanent I/O error is indicated, then ERP proceeds with error posting; if not, the action listed below for the appropriate command is performed. Except for the Dial command, the actions listed below apply only to a nonswitched line. If an error occurs on any command other than Dial, for a switched line, ERP indicates a permanent error and proceeds with error posting.
Dial (01)	Generate a channel program to perform a Disable, then TIC to the failing CCW.
Prepare (01)	Restart the channel program at the first CCW.
Poll (03)	Restart the channel program at the first CCW and start polling with the failing station.
Read Text (11) (Dynamic Buffering Not Used)	Set up special return code X'81' in IOBWORK+1 and generate a Read ENQ channel program. If this is a Read Initial operation on a multipoint line, set up for the Special Return routine to retry the Read using the second Read command (i.e., the one following the Read Index); if not a Read Initial on a multipoint line, set up for it to retry the Read using the failing CCW.
Read Text (11) (Dynamic Buffering Used)	Indicate a permanent I/O error, then proceed with error posting.
Write Text (11)	If the residual count is equal to the original count, restart the channel program at the failing CCW; if not, indicate a permanent I/O error, then proceed with error posting.
All other commands	Restart the channel program at the failing CCW.

Table 35. Data Check -- BSC

Command	Action
Read ENQ (0B)	If the retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, restart the channel program at the failing CCW.
Read Response to ENQ (0C)	If the retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, restart the channel program at the CCW preceding the failing CCW.
Read ID Response (07)	If the retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, restart the channel program at the preceding CCW if it is a Write ID ENQ CCW. If other than a Write ID ENQ CCW, restart at the failing CCW.
Read Response to Text (25)	If the retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, generate a channel program to Write ENQ, then TIC to the failing CCW.
Read Text (11) (Dynamic Buffering Not Used)	If the retry limit (7) has been reached, indicate a permanent I/O error, then proceed with error posting; if not, set up special return code X'8B' in IOBWORK+1 and generate a channel program to Write NAK, then TIC to the failing Read CCW. (TIC to the previous CCW if it is a Read response to text; otherwise, TIC to the Read text CCW.)
Read Text (11) Dynamic Buffering Used)	Indicate a permanent I/O error has occurred then proceed with error posting.
All other commands	Set "should not occur" bit in DECERRST, indicate a permanent I/O error has occurred, then proceed with error posting.

Table 36. Lost Data -- BSC

Command	Action
	Unless otherwise specified, each of the actions described below is preceded by a test to see whether the retry limit (7) has been reached. If so, a permanent I/O error is indicated, then ERP proceeds with error posting; if not, the action listed below for the appropriate command is performed.
Dial (01)	Generate a channel program to perform a Disable, then TIC to the failing CCW.
Prepare (01)	Restart the channel program at the failing CCW.
Read ENQ (0B)	If the CCW is part of a Write Reset channel program set the retry count to seven, indicate permanent I/O error, then proceed with error posting; if not, restart the channel program at the failing CCW.
Read ID Response (07)	Set up a special return code X'83' in IOBWORK+1 and generate a channel program to Read Response (with count =2).
Read Response to ENQ (0C)	Generate a channel program to Read Skip and set up for later restart of the channel program at the CCW preceding the failing CCW.
Read Response to text (25)	Set up special return code X'83' in IOBWORK+1 and generate a channel program to perform a Read Skip.
Read Text (11) (Dynamic Buffering Not Used)	Set up special return code X'82' in IOBWORK+1 and generate a channel program to Read ENQ (with count = 2). If this command is part of a Write Conversational channel program, set up for the special return routine to restart the channel program at the CCW preceding the failing CCW instead of at the failing CCW.
Read Text (11) (Dynamic Buffering Used)	The retry limit test is not performed. Set special return code X'1A' in IOBERRCT+1, set the retry count to seven, and generate a channel program to perform a Read Skip.
All other commands	The retry limit test is not performed. Set the "should not occur" bit (in DECERRST) and generate a channel program to perform a Read Skip. Set the retry count to seven and set special return code X'1A' in IOBERRCT+1.

Table 37. Timeout--BSC (Part 1 of 2)

Command	Action
	Unless otherwise specified, each of the actions described below is preceded by a test to see if the retry limit (7) has been reached. If so, a permanent I/O error is indicated, then ERP proceeds with error posting; if not, the action listed below for the appropriate command is performed.
Dial (01) Disable (01)	Restart the channel program at the failing CCW.
Read Response to EOT (0B) Read ENQ (0B)	<u>For a failing read ENQ command:</u> If the operation is a Read Initial (X'01') for a switched line, restart the channel program at the Read ENQ command; if a Read Initial for a nonswitched line, restart the channel program at the preceding command. If the operation is a Read Initial Inquiry (X'19') restart the channel program at the preceding command. If the operation is a Read Inquiry (X'15'), and no retries are requested, set the retry count to seven, then proceed with error posting. If it is a Read Inquiry and retries are requested, restart the channel program at the failing command. If the Read ENQ appears in an operation other than one of the foregoing, restart the channel program at the failing command <u>For a failing Read Response to EOT command, (The operation is a Write Reset):</u> Set the retry count to seven and proceed with error posting.
Read Response to ENQ (0C)	Restart the channel program at the CCW preceding the failing CCW.
Read Response to polling (0A)	Restart the channel program at the first CCW and start polling with the failing station.
Read Response to Addressing (06)	Restart the channel program at the first CCW.
Read ID Response (07)	If the maximum retry count <u>has not</u> been reached: If this is the calling station, restart the channel program at the CCW preceding the failing CCW; if it is the answering station, restart at the failing CCW. If the maximum retry count <u>has</u> been reached: If this is the calling station, set special return code X'8C' in IOBWORK+1 and Write DLE EOT, then disable the line; if this is the answering station, disable the line and TIC to the Enable CCW.

Table 37. Timeout--BSC (Part 2 of 2)

Command	Action
Read Text (11) (Dynamic Buffering Not Used)	If this is a polling operation, adjust CCW address and count (if necessary) to keep index byte location from being overlaid, set up return code X'81' in IOBWORK+1 and generate a channel program to Read ENQ (with count = 2); if not, restart the channel program at the failing CCW.
Read Text (11) (Dynamic Buffering Used)	The retry limit test is not performed. Indicate a permanent I/O error, then proceed with error posting.
Read Response to Text (25)	Set special return code X'86' in IOBWORK+1 and generate a channel program to write ENQ, then TIC to the failing CCW.
Write Transparent Text (11)	Generate a channel program to Write DLE ENQ, then TIC to the failing CCW,
All other commands	The retry limit test is not performed. Set "should not occur" bit (in DECERRST), indicate a permanent I/O error, then proceed with error posting.

Table 38. Unit Exception -- BSC (Part 1 of 3)

Command	Action
Write ENQ (03)	If NAK or RVI was received, set improper response (X'40') in DECFLAGS and restart the channel program at the CCW preceding the failing CCW. If neither NAK nor RVI was received, set special return code x'84' in IOBWORK+1 and generate a channel program to Read ENQ, with count=2.
Write Response to ENQ (08)	Set up special return code X'80' in IOBWORK+1 and generate a channel program to Read Response (with count=2).
Write Text (11)	Set special return code X'88' in IOBWORK+1 and generate a channel program to Read Response (with count=2).
Write Response to Text (08)	Set up special return code X'80' in IOBWORK+1 and generate a channel program to Read Response (with count=2).
Write EOT (21)	If the operation is a write reset, restart the channel program at the next CCW; if not, set special return code X'87' in IOBWORK+1 and generate a channel program to Read Response, with count=2.
Write WACK (01)	Set up special return code X'80' in IOBWORK+1 and generate a channel program to Read Response (with count=2).
All other Writes	Set "should-not-occur" bit (in DECERRST), indicate a permanent I/O error, then proceed with error posting.

Table 38. Unit Exception -- (Part 2 of 3)

Command	Action
Poll (03)	If the failing CCW is the first one in the channel program, set special return code X'88' in IOBWORK+1 and generate a channel program to Read ENQ (with count=2); if not, set up to start polling, beginning with the failing station, and restart the channel program at the first CCW.
Any Read command	If this is the first time through ERP, perform the action described below for the specific type of Read command; if not, turn off the 'ERP-in-control' indicator and restart the channel program. If two consecutive RVI's are received, proceed with error posting.
Read ENQ (0B)	If EOT was received, proceed with error posting; if not, determine whether the retry limit (7) has been reached. If it has, indicate a permanent I/O error, then proceed with error posting; if not, restart the channel program at the CCW preceding the failing CCW.
Read Response to ENQ (0C)	If the retry limit (7) has not been reached, and NAK or RVI was received, indicate improper response (X'40') in DECFLAGS and restart the channel program at the preceding CCW. If the retry limit has not been reached, and neither NAK nor RVI was received, set special return code X'84' in IOBWORK+1 and generate a channel program to Read ENQ, with count=2. If retry limit has been reached, indicate a permanent I/O error, then proceed with error posting.
Read ID Response (07)	If the operation is a Write Connect: If ID NAK or an invalid ID was received, restart the channel program at the CCW preceding the failing CCW. If after 7 retries ID NAK is still received, post the operation normally. If after 7 retries an invalid ID is still received, Write DLE EOT and disable the line. If the operation is a Read Connect: If an invalid ID was received, restart the channel program at the failing CCW, until the retry count of 7 is reached; thereafter, disable the line. For any other condition, or any other character received, set 'should-not-occur' bit (in DECERRST), indicate a permanent I/O error, then proceed with error posting.
Read Text (11)	1. If ENQ was not the last character received, set "should-not-occur" bit (in DECERRST, indicate a permanent I/O error, then proceed with error posting. 2. If ENQ was received and dynamic buffering is being used, proceed with error posting. 3. If ENQ was the only character received, dynamic buffering is not being used, the CCW preceding the failing CCW is a Write Response to Text (TP-Op code 08), and the retry limit (7) has not been reached: Restart the channel program at the CCW preceding the failing CCW (i.e., at the Write Response to Text CCW).

Table 38. Unit Exception -- BSC (Part 3 of 3)

Command	Action
	4. If ENQ was the only character received, dynamic buffering is not being used, the CCW preceding the failing CCW is a Read Response to Text (TP-Op Code 25), and the retry limit has not been reached: Generate a channel program to Write NAK and TIC to the CCW preceding the failing CCW (i.e., at the Read Response to Text CCW, which is part of a Write Conversational channel program). 5. If ENQ was the only character received, dynamic buffering is not being used, and the CCW preceding the failing CCW is neither a Write Response to Text (08) or a Read Response to Text (25): Set the "should-not-occur" bit (in DECERRST), indicate a permanent I/O error, then proceed with error posting. 6. If ENQ preceded by one or more characters was received, dynamic buffering is not being used, and the retry limit has not been reached: Generate a channel program to Write NAK and TIC to the failing CCW. 7. In 3, 4, and 6 above, if all conditions are met except that the retry limit has been reached, indicate a permanent I/O error, then proceed with error posting. 8. If the original count, minus one, does not equal the residual count and an SOH % message was received, proceed with error posting. 9. If STX ENQ was received and dynamic buffering was not specified, indicate X'01' in DECFLAGS, set special return code X'89' in IOBWORK+1, and generate a channel program to Write NAK and TIC to the failing CCW. If dynamic buffering was specified, proceed with error posting after indicating X'01' in DECFLAGS.
Read Response to Text (25)	If NAK was received: If dynamic buffering is used, proceed with error posting; if not, restart the channel program at the Write Text CCW. If ENQ was received: If the operation is a Write Inquiry, indicate contention (X'10') in DECFLAGS and restart the channel program at the failing CCW. If the operation is not a Write Inquiry, set special return code X'86' in IOBWORK+1 and generate a channel program to Write ENQ and TIC to the failing CCW. If neither ENQ nor NAK was received, set special return code X'86' in IOBWORK+1 and generate a channel program to Write ENQ and TIC to the failing CCW.
All other commands	Set "should-not-occur" bit (in DECERRST), indicate a permanent I/O error has occurred, then proceed with error posting.

Table 39. Special Return Code Actions (Part 1 of 6)

BTAM uses special return codes in two locations in the Input/Output Block (IOB) to cause ERP to perform the required functions in certain circumstances. The ERP control routine examines the code in IOBERRCT+1, and takes the actions shown in Part A of this table. The Special Return routine examines the code in IOBWORK+1 and takes the actions shown in Part B of this table.

Part A. Actions for special Return Codes in IOBERRCT+1

Code	Set:	Action
X'04'	When ERP writes DLE ENQ after detecting Command Reject.	If the Write DLE ENQ channel program completes normally, restart the user channel program at the Read Response to Text CCW that had previously ended with Unit Check and Command Reject. If the channel program completes with error, indicate a permanent I/O error, set the retry count to the maximum, 7, and proceed with error posting.
X'14'	(indicates that a special return code has been set in IOBWORK+1)	After the ERP-initiated channel program completes, the Special Return routine examines IOBWORK+1 to determine what further action to take, as shown in Part B of this table.
X'1A'	When ERP issues a Read Skip CCW after detecting a Lost Data condition, or when an error has occurred on an ERP Write CCW.	After the ERP-initiated channel program completes, ERP proceeds as follows: 1. If a timeout occurred following a Read Skip CCW, restart the channel program at the CCW that had ended with Unit Check and Lost Data indicated. 2. If an ERP CCW ends with sense bits other than Lost Data or Timeout on, set up to issue an I/O error message. 3. If an ERP Write CCW ends with an error, turn on the ERP-in-control indicator in the IOB, place code X'1A' in IOBWORK+1, and restart the channel program that had ended with the error that caused ERP to be initiated. 4. If an ERP CCW that is not a Write or a Read Skip ends with a Timeout, restart the ERP channel program at the beginning.
X'00'	1. Set before posting completion or returning to the supervisor. 2. Set when a Write DLE ENQ CCW was performed successfully after a Read Response to Text CCW ended with Command Reject (possibly indicating that the BSC adapter in the TCU was still in transparent mode).	The ERP control routine proceeds with its own analysis of the condition code, status, and sense information to determine the action to be taken, rather than being forced to pass control to a specific ERP routine.

Table 39. Special Return Codes (Part 2 of 6)

Part B. Actions for Special Return Codes in IOBWORK+1		
Code	Set:	Action
X'80'	When Equipment Check is detected on a Read Text CCW.	If ENQ was received: (for multipoint line) Clear the special return indicators, resend the last-sent acknowledgment, and TIC to the failing CCW. (for non-multipoint line) Clear the special return indicators and restart the channel program at the failing CCW. If ENQ was not received: (if retry limit has been reached) Clear the special return indicators, then proceed with error posting. (if retry limit has not been reached) Restart the ERP channel program at the beginning.
X'81'	1. When a Timeout on a Read Text CCW is detected (multipoint line). 2. When an Intervention Required error occurs on a Read Text CCW.	If ENQ was received: Clear the special return indicators and generate a channel program to Write NAK and TIC to the failing CCW. If ENQ was not received: (if retry limit has been reached) Clear the special return indicators and proceed with error posting. (if retry limit has not been reached) Restart the ERP channel program at the beginning.
X'82'	When Lost Data occurs on a Read Text CCW.	Clear special return indicators and generate a channel program to Write NAK and TIC to the failing CCW.
X'83'	When Lost Data occurs on a Read ID Response or Read Response to Text CCW.	Clear special return indicators and generate a channel program to Write ENQ and TIC to the failing CCW.

Table 39. Special Return Codes (Part 3 of 6)

Code	Set:	Action
X'84'	When Unit Exception is indicated on a Read response to ENQ or Write ENQ CCW.	If the operation is Write Inquiry (X'16'): If line is nonswitched, restart the channel program at the failing command. If line is switched: If ENQ was received, turn on the contention bit (bit 3) in DECFLAGS and clear the return indicators. If DLE EOT was received, proceed with error posting. If the operation is not Write Inquiry: If ENQ was received: If the operation is not Write Initial (X'02'), set "should-not-occur" bit (bit 1) in DECERRST, clear the special return indicators, and proceed with error posting. If the operation is a Write Initial: If the line is switched or multipoint, set "should-not-occur" bit (bit 1) in DECERRST, clear the special return indicators, and proceed with error posting. If the line is nonswitched: If this is not the primary station, turn on the contention bit (bit 3) in DECFLAGS, restore the CSW information in the CSW, clear the special return indicators, indicate no more retries are to be made, and return control to the supervisor. If this is the primary station and the retry limit has not been reached, restart the user channel program from the beginning. If this is the primary station and the retry limit has been reached, clear the special return indicators and proceed with error posting. If ENQ was not received: If the operation is Write Connect (X'1C), execute a Read Skip command. If the operation is not Write Connect (X'1C'), restart the channel program at the failing command.

Table 39. Special Return Codes (Part 4 of 6)

Code	Set:	Action
X'85'	(not used)	
X'86'	1. When Unit Exception has been indicated on a Read Response to Text CCW.	If a NAK is the only character received and dynamic buffering is being used: Clear the special return indicator and proceed with error posting. If a NAK is the only character received and dynamic buffering is not being used: Clear the special return indicator and restart the channel program at the Write Text CCW. If a NAK preceded by other characters is received: Clear the special return indicator and proceed with error posting. If the proper ACK (ACK-0 or ACK-1) is received: Clear the error indicators in IOBFLAG1 and the special return indicators, then return control to the supervisor. If something other than NAK, ACK-0, or ACK-1 is received: (If the retry limit (7) has been reached) Clear the special return indicators, then proceed with error posting. (If the retry limit has not been reached) Restart the ERP channel program. If the wrong ACK is received: (If dynamic buffering is being used) Clear the special return indicators and proceed with error posting. (If dynamic buffering is not being used) Clear the special return indicators and restart the channel program at the Write Text CCW.
	2. When timeout has occurred on a Read Response to Text.	If the failing CCW is a Write ENQ: If the maximum retry count (7) has not been reached, restart the channel program at the Write ENQ CCW. If the maximum count has been reached, proceed with error posting. If the correct alternating acknowledgment was received: post normal completion.

Table 39. Special Return codes (Part 5 of 6)

Code	Set:	Action
		If the wrong alternating acknowledgment was received: If dynamic buffering was used, proceed with error posting; if not used, and the failure occurred during a timeout situation, restart the channel program at the Write Text CCW; if not used and the failure did not occur during a timeout situation, resend the ENQ character. If NAK was received, restart the channel program at the Write Text CCW. If EOT or RVI was received, clear the error indicators in IOBFLAG1 and the special return indicators, then return control to the supervisor. If some character other than one of the foregoing was received, restart the channel program at the Write ENQ CCW.
X'87'	When Unit Exception is indicated on a Reset operation (TP-Op code 21)	If ENQ was received: Clear the special return indicators and restart the user channel program at the failing CCW. If ENQ was not received. (Failing CCW is the last CCW in user channel program) Restart channel program at failing CCW. (Failing CCW is not last CCW in user channel program) Restart channel program at the following CCW (Disable).
X'88'	When Unit Exception is indicated on a Poll or Write Text CCW.	Clear the special return indicators and restart the user channel program at the failing CCW.

Table 39. Special Return Codes (Part 6 of 6)

Code	Set:	Action
X'89'	When Unit Exception is indicated when STX ENQ is received.	If EOT was received, set the retry count to 7, clear the special return indicators, and proceed with error posting. If EOT was not received and the first characters are not STX or DLE STX, set "should-not-occur" bit in DECERRST. If the first characters are STX ENQ or DLE STX ENQ, and the maximum retry count has not been reached, restart the channel program at the failing CCW; if the retry count has been reached, proceed with error posting. If the first characters are not STX ENQ or DLE STX ENQ, clear the error indicators in IOBFLAG1 and the special return indicators, then return control to the supervisor.
X'8A'	When Unit Exception is indicate on a first ERP Write CCW.	Execute a Read Skip CCW and set special return code X'8A' in IOBWORK+1 after the original return code is saved. After executing the Read Skip, restore the original command and return codes.
X'8B'	When Data Check is indicated on a Read Text CCW (TP-Op code 11)	If ENQ was received, and the maximum retry count has not been reached, restart the ERP channel program (Write NAK and TIC to the Read CCW). If ENQ was not received, clear the special return codes and proceed with error posting.
X'8C'	When timeout is indicated on a Read ID Response CCW (TP-Op code 07)	Clear the special return indicators and proceed with error posting.

Table 40. Error Post Actions (1 of 2)

Condition	Action
Retry count is at limit (7)	Set up new polling or addressing characters, if applicable, and restore the original status and sense information to the IOB (this indicates the nature of the original error that occurred during the user channel program and that caused ERP to be initiated). If a Timeout error has occurred but the message is to be suppressed, pass control to the BTAM channel end appendage; if not, pass control to the operating system message writer.
Retry count is not at limit and:	
1. Failing CCW is not an ERP CCW	Same as for action when retry count is at the limit.
2. A special return code is present and the failing CCW is a Write CCW.	Set special return code X'1A' in IOBERRCT+1 to force control to be returned to the Error Post routine when the next interrupt occurs, indicate that ERP is in control (X'24' in IOBFLAG1), and restart the channel program at the CCW following the failing CCW.
3. Lost Data is indicated in the sense byte following execution of an ERP channel program for a Write Connect operation.	Indicate that ERP is in control (X'24' in IOBFLAG1), and restart the ERP channel program at the beginning.
4. An error other than Lost Data has occurred following execution of an ERP channel program for a Write Connect operation.	Indicate that ERP is in control (X'24' in IOBFLAG1) and restart the channel program at the CCW that ended with the error that caused ERP to be initiated.
5. An ERP Write CCW has ended with an error.	Same as action for condition 2, above.
6. The interrupt occurred on an ERP CCW other than Write, and no sense errors were indicated in the sense byte (ignoring Lost Data).	Return control to the ERP Control routine
7. The interrupt occurred on an ERP CCW other than Write, and the sense byte indicates that an error other than Timeout or Lost Data occurred.	Same as for action when retry count is at the limit.

Table 40. Error Post Actions (2 of 2)

Condition	Action
8. The interrupt occurred on an ERP CCW other than Write, the sense byte indicates Timeout and the Skip bit of the failing CCW is on.	Same action as for condition 4, above.
9. The interrupt occurred on an ERP CCW other than Write, the sense byte indicates Timeout, and the Skip bit of the failing CCW is _not_ on.	Restart the failing ERP channel program at the beginning.
10. SOH % E or SOH % C message was received.	Set up the fields used in printing operator-awareness messages generated by terminals for SYS1.LOGREC. The control is passed to the Teleprocessing Recorder.

ERRORS DETECTED BY THE DEVICE OR CONTROL UNIT AND CHANNEL DATA CHECK ERRORS

Error Conditions

Table 40A lists error conditions according to the bits that are on in the CSW and sense byte.

Recovery Actions

Table 40B indicates (by error condition and failing command) the recovery actions taken by BTAM error recovery procedures. If a diagnostic command is found in the failing CCW chain, action 1 is taken.

CSW Bits								Sense Bits								
32	34	35	36	37	38	39	44	0	1	2	3	4	5	6	7	Error Conditon
					X					X						1
					X				X							2
					X			X								3
	(X)		X	X	X					X						4
	(X)		X	X	X										X	5
	(X)		X	X	X							X				6
	(X)		X	X	X							X	X			7
	(X)		X	X	X									X		8
	(X)		X	X		X										9
	(X)		(X)	X		X										10
	(X)	(X)	(X)	X	X				X		X		X			11
	(X)	(X)	(X)	X	X				X							12
	(X)	(X)	(X)	X	X						X		X			13
	(X)	(X)	(X)	X	X							X				14
	(X)	(X)	(X)	X	X							X	X			15
	(X)	(X)	(X)	X	X									X		16
	(X)	(X)	(X)	X	X										X	17
X		(X)			X						X					18
X		(X)			X							X	X			19
							X									20

X indicates that the bit is on.

(X) indicates that the bit may be on if stacking is done by the channel.

Table 40A. Error Conditions (according to CSW and sense byte)

Failing Command	Command Code	Error condition (See Table 40A)																				
		1	2 CC≠3	2 CC=3	3	4[1]	5[1]	6[1]	7[1]	8[1]	9	10	11	12	13	14	15	16	17	18	19	20
Write	01	6[2]	4	1	2	3	2	3	3	3	7	2	1	4	1	3	3	3	2	6[2]	3	3
Erase write	05	6[2]	4	1	2	6[2]	2	6[2]	6[2]	6[2]	7	2	1	4	1	6[2]	6[2]	6[2]	2	6[2]	3	5
Diagnostic write	09	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	6[2]	3	1
Read buffer	02	6[2]	4	1	2	2	2	6[2]	3	6[2]	7	2	2	2	2	2	2	2	2	6[2]	3	6[2]
Read modified	06	6[2]	4	1	2	2	2	6[2]	3	6[2]	7	2	2	2	2	2	2	2	2	6[2]	3	6[2]
Diagnostic read	0A	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	6[2]	3	1
Select	0B	6[2]	4	1	2	2	2	6[2]	6[2]	6[2]	7	7	2	2	2	6[2]	3	6[2]	2	6[2]	3	5
Erase all unprotected	0F	6[2]	4	1	2	2	2	6[2]	6[2]	6[2]	7	7	2	2	2	6[2]	3	6[2]	2	6[2]	3	5
Sense	04	6[2]	4	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	6[2]	3	5
No operation	03	6[2]	4	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	6[2]	3	5

[1] This error condition may be the result of stacking in the channel.
[2] Action 3 is taken instead if it cannot be verified that the preceding command was not a write command.

Table 40B. Recovery Actions (by error condition and failing command)

The recovery actions are:

1. Permanent error
   The request is marked as a permanent error condition. The operation is terminated. The error is logged in the system error log. Unless the failing device is the console, a message is issued to the console operator indicating the failing device, operation, and conditions.

2. should not occur
   The request is marked as a nonrecoverable error condition. The operation is terminated.

3. Nonrecoverable error
   The request is marked as a nonrecoverable error condition. The operation is terminated. The error is logged in the system error log.

4. Assistance needed
   The request is held until a temporary device error condition is corrected. A message is issued to the cconsole operator asking for assistance. When the error condition is corrected, the operation is retried. The error is logged in the statistics table.

   Note: If the failing device is the console, action 3 is taken.

5. Retry failing CCW
   Restart the channel program on the failing CCW. When the retry count exceeds the number for the error condition and command, action 1 is taken.

6. Retry channel program
   Restart the channel program on the first CCW in the chain. When the retry count exceeds three, action 1 is taken.

   Note: If the channel program cannot be reconstructed or verified, action 3 is taken.

7. Busy
   The request is held until the device is ready. Then the operation is retried.

ERRORS DETECTED BY THE CHANNEL (EXCEPT CHANNEL DATA CHECK ERRORS)

Error Conditions

Table 40C lists error conditions according to the bits that are on in the ERPCODES field, which is byte seven of the Error Recovery Procedure Interface Block (ERPIB) built by the Channel Check Handler.

Note: Channel data checks are handled as though they were device-detected errors. Channel control checks and interface control checks are processed only if the Channel Check Handler is in the system.

ERPCODES Bits			Error Condition
5	6	7	
			21
		X	22
	X		23
	X	X	24
X			25
X		X	26
X	X		27
X	X	X	28
X indicates that the bit is on.			

Table 40C. Error Conditions (according to ERRCODES field in ERPIB)

Recovery Actions

Table 40D indicates (by error condition and failing command) the recovery actions taken by BTAM error recovery procedures. If a diagnostic command is found in the failing CCW chain, action 1 is taken.

The recovery actions are:

8. Permanent error
   The device is unable to recover. An ERP involving operator intervention may be given control, and the failing CCW may be retried.

9. Should not occur
   A message is issued to the console operator. Action 1 is taken.

10. Nonrecoverable error
    A message is issued to the console operator. Action 1 is taken.

11. Retry failing CCW
    The failing CCW is retried. If the error occurs a second time, a message is issued to the console operator, and action 1 is taken.

Failing Command	Command Code	Error Condition (See Table 40C)							
		21	22	23	24	25	26	27	28
Write	01	9	11	11	10	11	10	9	9
Erase write	05	9	11	11	10	11	10	9	10
Diagnostic write	09	9	8	8	8	8	8	9	8
Read buffer	02	9	11	11	11	11	11	9	11
Read modified	06	9	11	11	11	11	11	9	11
Diagnostic read	0A	9	8	8	8	8	8	9	8
Select	0B	9	11	11	11	11	11	9	11
Erase all unprotected	0F	9	11	11	11	11	11	9	11
Sense	04	9	11	11	11	11	11	9	11
No operation	03	9	11	11	11	11	11	9	11

Note: Action 10 is taken instead of action 11 if it cannot be verified that the preceding command was not a write command.

Table 40D. Recovery Actions (by error condition and failing command)

Once a user-program-issued Read or Write operation is completed, the program must decide what the next operation should be. This depends largely on the result of the preceding operation -- whether it was completed normally, with or without some exceptional condition or abnormally; and if the latter, what kind of error caused the abnormal completion. The tables in this chapter suggest, for various completion codes and ending conditions, the next READ or WRITE macro it might be appropriate for the user program to issue.

Table 41. Retry Options for Write Operations (Nonswitched Point-to-Point Line [BSC1])

TP-OP Code (hex)	Completion Code (hex)	Other Indications (hex)	Meaning	Retry Options
0C	7F	DECFLAGS: 40	NAK received in response to ENQ	1
	7F	DECFLAGS: C0	WACK received in response to ENQ	1,3, or 6
	41	ENQ in DECRESPN	ENQ received in response to ENQ (MODE=CNTRL in DCB macro)	1
	41	DECFLAGS: 10	ENQ received in response to ENQ (MODE =CNTRL in DCB macro)	5
11	41	Data check	(for TIV, TIVX, TTV, TTVX only) Text was received with error	2
25	7F	DECFLAGS: 20	Wrong acknowledgment received in response to text	3
	7F	DECFLAGS: 40	EOT received in response to text	1
	7F	DECFLAGS: 42	RVI received in response to text	7
	7F	DECFLAGS: C0	WACK received in response to text	3 or 6
	41	DECFLAGS: 40	NAK received in response to text	4

Retry Options:

1. Issue a WRITE Initial (TI) macro.
2. Issue a READ Repeat (TP) macro.
3. Issue a WRITE Inquiry (TQ) macro.
4. Issue a WRITE Continue (TT) macro.
5. Issue a READ Initial (TI) or READ Initial Inquiry (TIQ) macro.
6. Issue a WRITE Reset (TR) macro.
7. Continue normally.

Table 42. Retry Options for Read Operations--Answering (Switched Point-to-Point [BSC2])

TP-Op Code (hex)	Completion Code (hex)	Other Indications (hex)	Meaning	Retry Options
0B	41	Timeout	ENQ not received	1
	7F	DECFLAGS: 10	Received [ID] ENQ did not match expected [ID] ENQ	6
	41	Timeout	[ID] ENQ not received	5
11	41	Lost data, data check, or overrun	Text was received with error	2 or 4
	41	Timeout	No text received	1,3 or 4
	41	DECFLAGS: 01	STX ENQ received in lieu of text	4

Retry Options:

1. Issue a WRITE Break (TB) macro to disconnect the line.
2. Issue a READ Repeat (TP) macro.
3. Issue a READ Inquiry (TQ) macro.
4. Issue a WRITE Disconnect (TD) macro to disconnect the line.
5. If expanded ID verification is in use, BTAM automatically disconnects the line and reissues the READ Connect macro. If expanded ID verification is not in use, issue a WRITE Break (TB) macro.
6. If expanded ID verification is in use, BTAM automatically disconnects the line. You may therefore reissue the READ Connect macro. If expanded ID verification is not in use, issue a WRITE Break (TB) macro.

Table 43. Retry Options for Read Operations -- Calling (Switched Point-to-Point [BSC2])

TP-Op Code (hex)	Completion Code (hex)	Other Indications (hex)	Meaning	Retry Options
0C	41	Timeout	No response received to ENQ	1
0B	41	Timeout	ENQ not received	1
07	7F	DECFLAGS: 10	Invalid ID received in response to ID ENQ	2
	41	Timeout	No response received to [ID] ENQ	1
11	41	Lost data, data check, or overrun	Text was received with error	1 or 3
	41	Timeout	No text received	1,2, or 4

Retry Options:

1. Issue a WRITE Disconnect (TD) macro to disconnect the line.
2. Issue a WRITE Break (TB) macro to disconnect the line.
3. Issue a READ Repeat (TP) macro.
4. Issue a READ Inquiry (TQ) macro.

Table 44. Retry Options for Write Operations (Switched Point-to-Point Line [BSC2])

TP-Op Code (hex)	Completion Code (hex)	Other Indications (hex)	Meaning	Retry Options
0C	7F	DECFLAGS: 20	Wrong acknowledgment received in response to ENQ	1 or 4
	7F	DECFLAGS: 40	NAK received in response to ENQ	1 or 4
07	7F	DECFLAGS: 10	Invalid ID received in response to ID ENQ	If original operation is Write TI, option 4. If original operation is Write TC, option 5
	7F	DECFLAGS: 40	NAK or ID NAK received in response to ID ENQ	4 or 5
	7F	DECFLAGS: C0	WACK received in response to ID ENQ	4 or 5
	41	Timeout	No response received to [ID] ENQ	If original operation is Write TI, option 4 If original operation is Write TC, option 5
11	41	Lost data, data check, or overrun	(for TIV,TIVX,TTV, TTVX only) Text was received with error	3 or 4
25	7F	DECFLAGS: 20	Wrong acknowledgment received in response to text	1 or 4
	41	NAK in DECRESPN	NAK received in response to text	2 or 4
	41	Timeout	No response received to text	1 or 4

Retry Options:

1. Issue a WRITE Inquiry (TQ) macro.
2. Issue a WRITE Continue macro to retransmit the text.
3. Issue a READ Repeat (TP) macro.
4. Issue a WRITE Disconnect (TD) macro to disconnect the line.
5. Reissue the WRITE Connect (TC) macro.

TP-OP Code (hex)	Completion Code (hex)	Other Indications (hex)	Meaning	Retry Options
0A	41	Timeout	No index byte was received	1,2, or 4
09	7F	DECFLAGS: 04	Negative response to polling	1, 2, or 4
03	41	Timeout	No terminal responded to polling	1, 2, or 4
	48		Initial read terminated by RESETPL macro	1, 2, or 4
11	41	DECFLAGS: 01	STX ENQ sent in lieu of text	2
	41	Lost data, data check, or overrun	Text was received in error	2 or 3
	41	DECFLAGS: 40	Text was received ending with an ENQ	5
	7F	DECFLAGS: 02	Error status message was received	6

Retry Options:

1. Issue a READ Initial (TI) macro to poll the same or a different station.
2. Issue a WRITE Reset (TR) macro.
3. Issue a READ Repeat (TP) macro.
4. Issue a WRITE Initial (TI) macro.
5. Issue a READ Initial (TI) macro (using the polling entry of the remote 3270 device for which completion was posted) to receive the error status message.
6. Issue a READ Continue (TT) macro, and examine the sense/status bytes to determine what action to take.

Table 45. Retry Options for Read Operations (nonswitched Multipoint Lines (BSC3))

TP-OP Code (hex)	Completion Code (hex)	Other Indications (hex)	Meaning	Retry Options
06	7F	DECFLAGS: 04	NAK received in response to addressing	1, 2, or 7
	7F	DECFLAGS: 42	RVI received in response to addressing	1, 2, or 7
	7F	DECFLAGS: 42	RVI received (remote 3270)	2 or 8
	7F	DECFLAGS: C0	WACK received in response to addressing	1 or 2
	41	Timeout	No response received to addressing	1 or 2
11	41	Data Check	(TIV, TIVX, TTV, TTVX only) Text was received with error	2 or 5
	41	DECFLAGS: 40	(TIV only) Text was received ending with an ENQ	8
25	7F	DECFLAGS: 20	Wrong acknowledgment received in response to text	2 or 4
	7F	DECFLAGS: 42	RVI received in response to text	2 or 6
	7F	DECFLAGS: C0	WACK received in response to text	2 or 4
	7F	DECFLAGS: C0	WACK received (remote 3270)	9
	41	DECFLAGS: 40	NAK received in response to text	2 or 3
	41	DECFLAGS: 40 DECRESPN: EOT	EOT received in response to text	8
	41	Timeout	No response received to text	2 or 4

Retry Options:

1. Issue a WRITE Initial (TI) macro to address the same or a different station.
2. Issue a WRITE Reset (TR) macro to terminate selection.
3. Issue a WRITE Continue (TT) macro.
4. Issue a WRITE Inquiry (TQ) macro.
5. Issue a READ Repeat (TP) macro.
6. Continue normally.
7. Issue a READ Initial (TI) macro to poll another station.
8. Issue a READ Initial (TI) macro (using the polling entry of the remote 3270 device for which completion was posted) to receive the error status message.
9. If the write operation started a printer, issue a WRITE Reset (TR) macro to reset the line, and continue normally.

Table 46. Retry Options for Write Operations (Nonswitched Multipoint Line (BSC3))

Completion Code (hex)	Other Indications (hex)	Meaning	Retry Options
7F	DECFLAGS:01	OLTEP received control of the device following normal completion of the I/O operation	1, 2, 3, or 4
41	DECFLAGS:01 (and other error flags)	OLTEP received control of the device following a permanent I/O error	2, 3, or 5
41	DECERRST:80	Control unit not operational	1, 2, or 3
41	DECERRST:00 DECSENSO:01	Incorrect data stream	6
41	DECERRST:10	Integrity of the device regeneration buffer is questionable	4
41	Other than above	I/O error	5
44		I/O request intercepted	7
48		Read TI canceled	1

Retry Options:

1. Continue normally.
2. Inform the system operator, and request additional information.
3. Wait for some interval of time before trying the next I/O operation.
4. Issue a WRITE TS macro instruction to reconstruct the buffer contents.
5. Further use of the device is questionable, although not prohibited. The problem program should consider the device unavailable and should consider requesting that diagnostics be run on the device.
6. Check that the data stream is correct (that is, buffer addresses are correct, order sequences are complete, and orders do not cause overrun).
7. The contents of the device buffer are doubtful, because (1) an error occurred following the completion of the previous I/O operation or (2) a request-for-test message was received from the device requesting that a test message be sent to another device. If the current operation is a write erase, it should be done. Otherwise, a WRITE TS macro instruction should be issued to reconstruct the buffer contents before doing the current operation.

Table 46A. Retry Options for Local 3270 Read and Write Operations

On-line testing is an optional BTAM facility that permits the user to verify proper operation of terminals and of the communication lines that link them to the computer, and to aid in diagnosing line or terminal troubles. On-line testing centers around transmission of predefined standard test messages, the formats of which depend on the purpose of the test.

On-line testing is performed during normal BTAM operation. Only the communication lines and terminals specified are involved; data transmission proceeds as usual on other lines. Operation of the program is affected only to the extent of the line time required for test transmissions and of the CPU time required to process requests for tests.

You may wish to perform certain kinds of on-line testing as a routine procedure, for example, to test line or terminal functioning at the beginning of each day, or at intervals during the day. Other kinds of tests are appropriate as diagnostic aids, and are normally performed as needed by the computer or terminal operator, or IBM customer engineer.

In order to have the on-line testing facility available, you must code T among the EROPT operands of the DCB macro instruction for the line group.

On-line testing is implemented somewhat differently for start-stop lines and for binary synchronous lines.

ON-LINE TESTING FOR START-STOP COMMUNICATIONS LINES

For start-stop communication lines, test requests may be initiated only at remote terminals. The tests requested may involve message switching, comparing the contents of a test message to a predefined character sequence in main storage, sending a string of characters to a specified terminal, or checking the IBM SELECTRIC typing element mechanism of a terminal printer.

Start-stop on-line tests are initiated by transmission of test request messages, the format of which is:

99999	xx	type	TO ADDR	UNIT SELECT	text	END CHAR
5	2	1	1or2	1or2	Variable	1

Field length (bytes)

99999
: identifies this message as a test request.

xx
: defines the type of test to be executed (see Test Type Codes, below).

type
: specifies the type of terminal from which the test is being requested. Applicable type codes are:

Code	Device
0	2741
1	1030 card reader
2	1050
3	1060
4	2740 (with or without 2760 attached)
5	1030 badge reader or manual entry unit.
6	2260 (Remote) and 2265 (Remote)

TO ADDR
: specifies the address of the terminal to which the message is to be sent (for 2760 tests, specifies function to be performed).

TO ADDR is a one-byte field for the IBM 1030 Card Reader and 1050, 2740, and 2741 terminals; it contains the addressing character for the selected terminal. For those 2740 and 2741 terminals not using addressing characters (i.e., all terminals not equipped with station control), this field should contain a space character (in the hexadecimal representation of the transmission code pattern for space) except when a 2760 frame change or scan point test is performed.

TO ADDR is a two-byte field for 1030 badge readers and manual entry units, 1060 terminals, 2260 and 2265 terminals; it contains a two-byte code indicating which addressing characters BTAM is to send on the line.

For 1030

Code	Addressing Character
02	B
03	C
04	D
.	.
.	.
.	.
26	Z

Note: Codes of 01 and 10, representing A and J, may not be used, as A and J are invalid 1030 addresses.

For 1060

Code	Addressing Character
01	A
02	B
03	C
.	.
.	.
.	.
26	Z

For 2760

TO ADDR contains the F-character that specifies the 2760 function to be performed. Figure 14 lists the F-characters and their meanings.

For 2848 (2260) and 2845 (2265)

TO ADDR is used to select the 2848 or 2845 display control unit. The address of a display control unit can be any USASCII non-control character (i.e., any character in columns 3-7 in the USASCII code chart), therefore allowing 96 possible display control addresses.

Actual Unit Address $b_7 \ldots b_1$	Code
0100000	01
0100001	02
...	...
1111111	96

Note: The TO ADDR code applicable to a particular display control unit can be determined from one of its attached display stations by specifying the Request Address test (test type 09) in the test message.

UNIT SELECT

Note: Unit select is not applicable to 1030, 2740, or 2741 tests; therefore, text can start in this position.

For 1050 and 1060 (1 Character)

UNIT SELECT specifies the particular component of the selected terminal that is to receive the message, i.e., 1052, 1053, 1055, 1062 Printer 1 or 2, etc. The appropriate unit select code can be determined from the SRL publication pertaining to the terminal. For 2760 tests, this field contains the A_1, A_2 characters that specify amount of filmstrip movement. See Figure 8.

For 2260 or 2265 (or 1053 Attached to the 2848 or 2845) (2 characters)

2260 and 2265 Display Stations and 1053 Printers are selected by transmitting a predefined code in these character positions. The device selection code can be one of 25 USASCII non-control characters.

Actual Unit Address $b_7 \ldots b_1$		Code
1000000	2260 and 1053 attached to 2848	01
1000001		02
...		...
1011000		25
1011001 -	2265 attached to 2845	26
1011001 -	1053 attached to 2845	27

Note: The UNIT SELECT code applicable to a particular 2260 display station can be determined from that display station itself by utilizing the Request Address test (test type 09).

END CHARACTER

1030 = EOB
1050 = EOT
1060 = EOB
2740 = EOT
2741 = EOT
2760 = EOT
2848 = ETX

Note: The test message is transmitted from a 1060 terminal by utilizing the data and transaction keys. The EOB character is entered by depressing the teller A or B key.

TEST TYPE CODES

01 Message Switching
This test receives a message from the requesting terminal and transmits it to the terminal (on the same line) specified in the test message. Note: The length of the message to be switched cannot exceed the length of the data area specified in the READ macro for the line over which the test is requested.

02 Tilt
This test sends the tilt test to the requested terminal. This test is designed to check the SELECTRIC typewriter print ball mechanism.

03 Rotate
This test sends the rotate test to the requested terminal. This test is designed to check the SELECTRIC typewriter print ball mechanism.

04 Twist
This test sends the twist test to the requested terminal. This test is designed to check the SELECTRIC typewriter print ball mechanism.

05 Stored Compare
This test provides a means to compare the received message with a particular character sequence in main storage. The message in main storage is compatible with the transmitting capabilities of the terminals involved.

The test message to be compared with the character sequence in main storage is transmitted from the terminal and consists of the numbers 0 through 9 followed by the alphabet (A through Z). The incoming test message must specify the comparison characters in the same order as they appear in the sequence in main storage although not all of them need be specified.

The length of the test message cannot exceed the length of the data area specified in the READ macro that will receive the message. The data area must be long enough to contain the header information (99999, etc.), the characters to be compared, and the end character.

Exceptions:

1. When transmitting from any 2740 terminal, a space character must precede the comparison data. This space character is in addition to the space character in the TO ADDR field.

2. The stored compare test for a 1060 is requested by entering the following message:

```
┌───────────────────────────┐
│9 9 9 9 9 6 5 3 4 2 1 0 EOB│
└───────────────────────────┘
```

Comparison is then made to this message. Responses to this request are printed only at the requesting terminal.

Messages received at the terminal are:

1. If the comparison to the stored message is valid, the following message is sent to the terminal specified in the TO ADDR field:

```
┌───────────┐
│ CMP VLD-* │
└───────────┘
```

The character printed in the position of the asterisk will be the last character against which a comparison could be made. Exception: The message sent to a 1060 after a valid comparison is:

```
┌───────────┐
│ CMP VLD   │
└───────────┘
```

If the request was received properly, but an insufficient count was specified in the READ and thus no characters could be compared, a / character is printed in the asterisk position.

2. If the comparison to the stored message is invalid, the data received is message-switched to the terminal specified in the TO ADDR field.

Note: The Stored Compare test is not applicable for the 1030 manual entry unit or badge reader.

06 All Characters Test
This test provides the standard All Characters test for IBM Customer Engineer terminal checkout and serves as a start-up message. Special characters are not used in the terminal test. Characters received at the terminal are:

For 1030, 1060, 2848 (2260 and 1053):

Numbers: 0-9, and alphabet: A-Z.

For 1050, 2740, 2741:

Numbers: 0-9, alphabet a-z (lower case), and alphabet A-Z (upper case).

07 SELECTRIC Analyzer Test

This test provides an exercise to analyze the capability of the SELECTRIC typewriter carrier mechanism to perform within specifications. When this test is requested, BTAM sends to the terminal a predefined message that exercises the carrier mechanism. This test is not applicable to a 1053 Printer attached to a 2848 or 2845 Display Control.

08 Write at Line Address Test (2260 and 2265)

This test provides line selectivity checkout by using the first two characters after the UNIT SELECT field as a new display line code. This can be followed by data which is to be switched to the terminal and displayed on the display station screen at the selected line. The codes and associated display lines are:

Code	Display Line
01	1
02	2
03	3
.	.
.	.
12	12

09 Request Address Test (2260 and 2265)

This test allows the operator at a display station to determine the display control and display station address applicable to that station.

The TO ADDR and UNIT SELECT fields are not utilized in this test message since the test itself provides these fields to the requesting terminal. ETX can be sent immediately after the TYPE field.

BTAM returns to the requesting display station a 9 character message giving the addressing information for that station. The format is:

DC+DVxxyy

DC+DV indicates that the message contains the requested addressing information; xx and yy are the display control and device (i.e., display station) addresses.

Note: This test provides only the TO ADDR and UNIT SELECT codes of the requesting display station. It is not a means of getting these codes for some other display station.

10 Frame Change Test (2760)

This test enables an IBM Customer Engineer to request that a filmstrip be moved to a new frame. The request for a frame change test is entered on the 2740 keyboard. BTAM uses the data in this message to generate the appropriate frame change message and sends it to the 2760. The Customer Engineer visually verifies the correctness of the film movement.

11 Scan Point Test (2760)

This test performs a filmstrip movement and then allows the Customer Engineer to probe the screen and have the horizontal and vertical coordinates of the probed response points printed on the 2740 printer. The request for a scan point test is entered at the 2740 keyboard. BTAM generates a frame change message and sends it to the 2760. The Customer Engineer then probes one or more response points, depending on the mode specified in the test request message. BTAM sends to the 2740 a message containing the coordinates of the response points probed.

See On-Line Testing under IBM 2760 Optical Image Unit - General Information, for further information on 2760 on-line tests.

TERMINAL TEST RESTRICTIONS

1. A remote terminal may send a test request message only when the operation in effect for the line is a Read Initial or Read Conversational operation.

2. The user program input area must be long enough to accommodate the entire test message. The response to polling must be read into the first byte of this area. If dynamic buffering is used there is an additional restriction: the data area of the first buffer in the chain must contain all of the characters in the test request.

3. No READ macro that specifies an answering list can include the Reset option. For example, a READ TI or TV can be issued, but not a READ TIR or TVR , for a line over which test requests may be received. The line connection must be maintained during the terminal test (the Reset option causes BTAM to break the connection).

4. To request a test from a 1030 badge reader, the badge reader must be wired to read out the entire 10 columns of the badge (refer to SRL publications for the IBM 1030).

5. The transaction code received from a 1030 is not included as part of the test request.

6. All 1030 tests require a 1033 Printer on the same line as the requesting terminal. The printer address must be specified in the TO ADDR field.

7. The terminal tests will not test 1035 Badge Readers or 1030 Badge Readers in a 1035 environment.

8. If insufficient storage is available for the test pattern, the request will be switched to the terminal specified by the TO ADDR field.

ON-LINE TESTING FOR BINARY SYNCHRONOUS COMMUNICATIONS LINES

On-line tests for BSC lines may be requested by the central computer, by remote stations, or both, depending upon the type of test and the line and station configuration. There are 23 types of tests, not all of which apply to all configurations.

On-line testing is available for all types of remote BSC stations. For S/360-to-S/360 operation, both computers may run under BTAM with the on-line test facility, or one may run under BTAM and the other under an on-line diagnostic program. Operation between S/360 and a S/360 Model 20, 1800, 2715, 2770, 2780, 2972, or remote 3270 requires the S/360 to run under BTAM or an on-line diagnostic program. For S/360 to System/3 or 1130 operation, the S/360 must run under BTAM, and the System/3 or 1130 must run under an on-line diagnostic program.

In S/360-to-S/360 operation, either computer may initiate on-line tests. In operations between the central computer and a S/360 Model 20, System/3, 1130, 1800, 2770, or 2972, the central computer cannot initiate the on-line test except for a test type of 0. In operations between the central computer and a 2715, only the 2715 can initiate an on-line test. In operations between the central computer and a remote 3270, any remote terminal on the same line can initiate an on-line test of the remote 3270.

When the central computer initiates the test with a 2780, the 2780 mode switch must be set to either Print or Punch position if the 2780 is on a point-to-point line.

Tests are requested at a remote station by sending to the central computer a message having a special format, called a <u>request-for-test</u> (RFT) message. The method of sending the RFT message differs for the various types of remote station. For a 2780, the RFT message is punched in a card. For an 1130, System/3 or System/360 Model 20, the message is sent by a diagnostic program. For a remote 3270, (1) the cursor is positioned at the top left of an unformatted screen (by pressing the CLEAR key and then the RESET key, for example), (2) the text of the RFT message (test type, number of times, length of address, selection address) is entered by means of the keyboard, and (3) the TEST REQUEST key is pressed to frame the text with control characters (SOH % / STX and ETX).

To request a test at the central computer, the programmer codes an ONLTST macro instruction in the program at the point at which the test is to be performed. The ONLTST macro generates the proper RFT message and sends it to the remote computer or terminal specified in the ONLTST macro.

The format of the RFT message is the same whether it is sent by the central computer or is received by the central computer from a remote computer or terminal. The format is shown below under Formats of RFT, Test, and Console Messages.

Transmission of an RFT message is followed by one or more transmissions of test messages. The RFT message contains a field called the X field, which contains a code indicating the type of test to be performed. The code, from 00 to 22, governs the sequence of I/O operations comprising the test and determines the content of the test message.

TYPES OF TESTS

<u>Type 00</u>

For this type of test the requesting station sends an RFT message, immediately followed by a test message, or a sequence of test messages, the content of which is user-specified. The test message is sent the number of times specified in the Y field of the RFT message, which may be from one to 99. For example, if you specify a Y value of 5, the requesting station sends the RFT message, followed by five consecutive transmissions of the same test message. The computer or terminal that receives the RFT and test message responds with an acknowledgment after each message.

For this type of test, the requesting station may be the central computer except when the remote station is a 2715 Model 1, or any type of remote station. When the requesting station is a 2770,2780, or 2972, however, the operation differs somewhat. First, the test message is sent not as a separate message following the RFT message, but as a part of the RFT message itself. Second, the Y field of the RFT message can only be coded as one, since the RFT message, including the message text, is sent only once.

Another restriction applies when the station receiving the RFT message is a 2770 or 2780. The job switch (2770) or mode switch (2780) must be set to permit the RFT message to be received at the printer, card punch, paper tape punch (2770), or display (2770), unless the text contains component selection characters.

The requesting station may not be a remote 3270 display station.

Type 01

For this type of test, the requesting station sends an RFT message that includes user-specified text characters. The station receiving the RFT message acknowledges it, prepares a test message containing the text characters from the RFT message, and sends the test message the number of times specified in the Y-field of the RFT message -- from 1 to 99. The station receiving the test messages (i.e., the station that sent the test request) responds with an acknowledgment after each test message.

For this type of test, the requesting station may be the central computer only if the remote station is a System/360 (excluding Model 20). The requesting station may be any type of remote station. If the requesting station is a 2770 or 2780, its job switch (2770 or mode switch (2780)) must be set to permit the test messages returned from the central computer to be received at the printer, card punch, paper tape punch (2770), or display (2770), unless the text contains component selection characters.

Notice that in type 00 tests, the requesting station also sends the test messages, and receives acknowledgments in reply, while in type 01 tests the requesting station receives test messages in reply.

Types 02-34

For these types of tests, the requesting station sends an RFT message. Unlike tests of types 00 and 01, the RFT message neither contains nor is followed by a test message. Instead, the X field of the RFT message indicates to the receiving station which of 34 BTAM-defined standard test messages it is to return to the requesting station. When BTAM receives the RFT message, it examines the X and Y fields, selects the test message designated by X, and sends it Y times. The contents of test messages for each type of test are given below under Formats of RFT, Test, and Console Messages.

For this type of test, the requesting station may be the central computer only if the remote station is a System/360 using BTAM. The requesting station may also be any type of remote station. If the requesting station is a 2770 or 2780, its job switch (2770) or mode switch (2780) must be set to permit the test messages sent from the central computer to be received at the printer, card punch, paper tape punch (2770), or display (2770), unless the text contains component selection characters.

Note: Set the 2780 'On Line Test' switch to the on position. This will suppress the generation of an STX character preceding the RFT message.

BTAM RESPONSES TO REQUEST-FOR-TEST MESSAGES

BTAM recognizes and responds to any RFT messages received from a remote computer or terminal provided that:

1. The on-line test facility is available (you have coded T among the EROPT options in the DCB macro for the line group).

2. The RFT message was received on a Read Initial (TI) operation. Or, if the device to be tested is part of a remote 3270 display system, the RFT message may have been received on a Read Continue (TT) operation.

3. The length of the input area specified by the READ macro is at least 300 bytes for test types 02-34. If buffering is used, the entire 300-byte area must be contained wihtin on buffer. For test types 02-34, if the area is less than 300 bytes, BTAM returns and EOT instead of a test message. The EOT ends the test before any test messages are sent. For test types 00 and 01, no check is made to determine the length of the input area; instead the length specified in the READ macro is used. You must ensure that the area is large enough to accommodate the text data in the RFT message or the test message that follows the RFT message. Otherwise lost data and timeout errors will result.

4. The RFT message was received without error.

The remote computer or terminal may send an RFT message only when the BTAM program has a Read Initial operation pending on the line over which the RFT message will be received, unless the device to be tested is part of a remote 3270 display system, which can send an RFT message on a Read Continue operation.

When BTAM recognizes the message received by a Read Initial operation as an RFT message, the Read operation is not posted complete as it is for non-RFT messages. Instead, control is given to the on-line test logic, which examines the RFT message, generates the requested test message in the area specified in the READ macro, and sends the test message to the requesting computer or terminal (or other specified destination, for multipoint lines). If the RFT message specified a type 00 test, only a response is returned to the requesting computer or terminal, as explained previously. Following transmission of the test message the requested number of times, the on-line test logic sends an EOT character for nonswitched lines, or DLE EOT (and disables the line) for switched lines, then restarts the program at the Read Initial operation that received the RFT message. When an RFT message is received for a remote 3270 display station on a Read Continue operation, BTAM gives control to the on-line test logic, which generates and sends the test message and then posts the Read Continue operation complete and places an EOT in the input area specified in the read operation.

Notes:

1. Test mode will not be entered until the RFT message is received correctly and positively acknowledged and until the proper positive response (ACK-0) to selection or line bid is received. If a positive response to selection (ACK-0) is not received initially or after seven retries, the test will be terminated.

2. Once test mode has been entered, if one or more WACK responses are received, the transmitting station will respond to each WACK with an ENQ, until the regular positive response is received. The number of WACKs that will be accepted is 25; if more than this number are received consecutively, the on-line test is terminated.

3. When a test message is requested for a remote 3284 or 3286 printer, the RFT message should specify that the test message be sent only once. This avoids wasting line time, since the test message appears only once on a remote 3270 printer even though attempts are made to send it more than once when the Y field of the RFT message is greater than one.

BTAM INITIATION OF REQUEST-FOR-TEST MESSAGES

As mentioned previously, you may initiate on-line tests by coding the ONLTST macro instruction in your program. ONLTST causes the on-line test logic to prepare an RFT message, send it, send or receive test messages (depending on test type), receive or send appropriate acknowledgments, and accumulate and display on the central computer console the results of the test. The ONLTST macro is described below. Message formats for each type of test are given under Formats of RFT, Test, and Console messages.

ONLTST (On-Line Test) Macro Instruction

The ONLTST macro instruction is used to send a request-for-test (RFT) message on a binary synchronous communication line. It provides the information necessary to build the RFT message, generates the linkage to the on-line test routine, and causes the RFT message to be sent.

The Write operation executed by the ONLTST macro is similar to a Write Initial operation; the ONLTST macro must therefore be used in the same manner. That is, it may appear in your program only where a Write Initial macro could appear. ONLTST may be issued only when the computer or terminal that is to receive the RFT message is capable of recognizing it as such and acting accordingly. For example, if the computer that is to receive the RFT message is operating under BTAM, ONLTST may be issued only when the corresponding operation at the receiving computer is a Read Initial or Read Connect operation for which the input area length is at least 300 bytes.

Upon completion of an on-line test on a switched point-to-point line, BTAM breaks the line connection.

After issuing an ONLTST macro, you must issue a WAIT or TWAIT macro (or otherwise test for completion of the on-line test) before starting any other Read or Write operation for the line.

After execution of the ONLTST macro, control is returned to the next sequential instruction in the user program.

Note: ONLTST cannot be issued to initiate a test between the central computer and an IBM 2715 or a remote IBM 3270.

Name	Operation	Operands
[symbol]	ONLTST	DECB=decb address, X=type of test, Y=no. of transmissions, DCB=dcb address, AREA=rft message area [,TEXT=user text area, LENGTH=user text length] [,ENTRY=list address] [,RLN=line number]

DECB
: specifies the address of the data event control block for the line on which the on-line test is to be performed.

X
: specifies the type of test to be performed. Permissible values of X and their meanings are tabulated below, under Formats of RFT, Test, and Console Messages.

Y
: specifies the number of times the test message is to be transmitted. Y may be from 1 to 99.

DCB
: specifies the address of the data control block for the line group.

AREA
: specifies the address of the area from which the RFT message is to be sent. The on-line test routine formats the RFT message in this area and also reads into it the responding test messages, for test types 01-19. For type 00, BTAM moves the data comprising the test message into this area. For test type 00 or 01, this area must be large enough to receive the expected test message. For test types 02-22, this area must be at least 300 bytes long. If buffering is used, the entire area must be contained within one buffer.

TEXT
: specifies the address of the user-defined test message where X (test type) equals 0 or 1. For non-transparent text, you must begin and end the text with the appropriate framing characters (STX and ETX); for transparent text, you supply only DLE STX at the beginning of the text; BTAM provides the DLE ETX at the end of the message. Some amount of text data must be specified when the X operand is 0 or 1. For other values of X, this operand is not required, and is ignored if coded. The contents of this area are not destroyed by ONLTST and may be used for successive tests.

LENGTH
: specifies the number of text characters in the RFT message, where TEXT is specified. This operand must be coded if the TEXT operand is coded.

ENTRY
: specifies the address of the addressing or I.D. list (OPENLST, DIALST, or BSCLST types). The list must contain only one entry. A calling list, not an answering list, must be specified if the line is switched point-to-point.

  This operand is not used for on-line tests on point-to-point lines.

RLN
: specifies the relative line number of the line within the line group on which the test is to be performed.

Notes:

1. No ONLTST macro may be issued for a line until a data event control block has been established for that line by means of a READ or WRITE macro in list or standard format.

2. In an on-line test between a System/360 and a 2770, test messages sent to the 2772 control unit cannot exceed a length of 128 bytes, unless the 2772 has the Expanded Buffer feature, in which case the maximum length is 256 bytes.

Return codes: After an ONLTST macro is issued, BTAM sets register 15 to zero if no error was detected. If an abnormal condition is detected, the on-line test operation is not started, and control is returned to your program at the instruction following the ONLTST macro. A return code in register 15 indicates the error. Bits 0 through 23 are zero; bits 24 through 31 contain one of the following error codes in hexadecimal notation. (Code 0C is issued for the ONLTST macro itself; the other codes result from errors occurring when the on-line test routine executes a Write operation.)

04 Busy. The specified line is busy with a previously requested Read or Write operation.

08 Invalid RLN. The relative line number specified in ONLTST is zero or exceeds the number of lines in the line group.

0C Invalid test type or transmission count. The value specified by the X operand is undefined, or the value specified by the Y operand exceeds 99.

10 The skip bit of the addressing list entry specified by the ENTRY operand is on.

14 A line error occurred during Open.

18 On-line test facility was not specified in the EROPT operand of the DCB macro.

Note: All nonzero return codes indicate that no I/O operation was initiated; therefore the program must not issue a WAIT or TWAIT macro for an ONLTST macro that resulted in a nonzero return code.

Completion Codes: On completion of an on-line test operation, a completion code is set in the high-order byte of the event control block for the line being tested. The code, in hexadecimal notation, indicates the nature of the completion:

7F Normal completion: Channel end - Device end.

41 Operation completed with I/O error. The DECB for the line does not contain error indicators when this occurs. The operator at the computer executing BTAM receives a message indicating the nature of the error. It is suggested that the user program check the completion code and if it is 41, issue a Write-to-operator-with-reply (WTOR) macro to permit the operator to determine what further action should be performed (e.g., retry the on-line test by reissuing the ONLTST macro, or indicate to the user program that no further Read or Write operations can be performed on that line). In the latter case the operator can, after the error condition has been cleared, notify the program that I/O operations may be resumed.)

FORMATS OF RFT, TEST, AND CONSOLE MESSAGES

Request-For-Test Messages

An RFT message has one of two formats.

For type 00 tests[1]:

SOH %	X	Y	N	ADDR	STX ETX
2	2	2	1	0-9	2

Field length (bytes)

For type 01 tests[2] and type 02-34 tests requested from a station other than a remote 3270:

SOH %	X	Y	N	ADDR	Text
2	2	2	1	0-9	variable

Field length (bytes)

For type 02-34 tests requested from a remote 3270:

SOH %	/	STX	X	Y	N	ADDR	ETX
2	1	1	2	2	1	4	1

SOH %

/ identifies the message as an RFT message from a remote 3270.

X
: specifies the test type (00-22). X is a two-byte zoned decimal field.

Y
: specifies the number of times (1-99) the test message is to be sent. Y is a two-byte zoned decimal field. If X equals 0, and the remote station is a 2770, 2780, or 2972, Y must equal 1, because these stations transmit only the RFT message, not separate test messages. If the test message is to be set to a remote 3284 or 3286 printer, Y should equal one, since the test message appears only once on a remote 3270 printer.

N
: specifies the length (0-9) of the ADDR field. Code N as 0 and omit the ADDR field for tests over point-to-point lines, unless component selection characters are desired in the ADDR field.

ADDR
: Contains the address of the station or device to which the test message is to be sent, or (for 2770), component selection characters (DC1, DC2, or DC3). For a multipoint configuration, the ADDR field contains the selection address of the unit to which the test message is to be sent. (For the remote 3270, for example, //AA would be entered as the selection address for device 1 on control unit 1.) This need not be the same unit that sent the RFT message. For a point-

---

[1]except for RFT messages from a 2770 or 2780.

[2]and for type 00 RFT messages from a 2770 or 2780.

to-point configuration, the ADDR field contains the required component selection sequence, e.g., ESC x, where x indicates the component to be selected. This sequence is limited to two characters. The ADDR field is not present if N=0.

Text
: is the data and framing characters to be sent when X (test type) equals 00 or 01. For nontransparent text the data characters must be framed by STX or ETX. For transparent text the data characters must be framed by DLE STX and DLE ETX.

Test Messages

The contents of test messages are determined by the X field (test type) of the RFT message that initiates transmission of the test message. The values of X, the contents of the corresponding test message, and the configurations for which the test types are valid, are as follows:

X=00 For this test type, the test message is sent Y times, except for an RFT message from a 2770 or 2780, in which case the text is sent as part of the RFT message, not separately (RFT messages from a 2770 or 2780 must specify a Y value of 1). The RFT and test messages are acknowledged by DLE, ACK-1 if received without errors, by NAK if a data check is detected, and are not responded to at all if any other ending condition is detected. The RFT message and the following test messages (or included text data) can be received from any type of remote BSC station: S/360 (including Model 20), System/3, 1130, 1800, 2715, 2770, 2780, and 2972.

Note: If this test type is specified, the size of the input area specified by the Read Initial operation that receives the RFT message (via the DECB length parameter) must be large enough to receive the entire RFT message, including the text portion.

X=01 For this test type, the content of the test message is identical to the text portion of the RFT message, including the framing characters. The text is transmitted Y times. This message may be sent to any type of remote BSC station: S/360 (including Model 20), System/3, 1130, 1800, 2715, 2770, 2780, and 2972.

For the remaining test types, the text of the test message is predefined by the on-line test routine.

X=02 Transparent EBCDIC Message:

DLE STX ...Text... DLE ETX

The text consists of all 256 EBCDIC codes in collating sequence order. This message may be sent to a S/360 (including Model 20), System/3, 1130, 1800, 2715, and 2770.

X=03 Transparent USASCII Message:

DLE STX ...Text,.. DLE ETX

The text is in USASCII code (high-order bit always zero), and consists of all 128 USASCII codes in collating sequence order. This message may be sent only to a S/360 (including Model 20).

X=04 Normal EBCDIC Message:

STX SYN SYN ...Text... ETX

The text is in EBCDIC code, and consists of the 245 non-data link control characters. The characters excluded are SOH, STX, ETX, ETB, EOT, ENQ, ACK, NAK, SYN, US, DLE. This message may be sent to a S/360 (including Model 20), System/3, 1800, and 2770.[1]

X=05 Normal USASCII Message:

STX SYN SYN ...Text... ETX

The text is in USASCII code and consists of the 117 non-data link control characters. The excluded characters are the same as for X=04. (S/360-S/360, This message may be sent to a S/360, (including Model 20), System/3, 1800, 2770, and 2780.[1]

X=06 Alphameric USASCII Message:

STX SYN SYN A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 ETX

This message may be sent to a S/360, (including Model 20), System/3, 1800, 2770, and 2780.

X=07 USASCII Printer Message:

STX ESC Q A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 ETX

This message is used to test the IBM 2780 printer. It may also be sent to a S/360 (including Model 20), and an 1800; these stations treat the ESC Q sequence (printer selection code) as data.

X=08 USASCII Punch Message:

STX ESC 4 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 ETX

This message is used to test the IBM 2780 card punch. It may also be sent to a S/360 (including Model 20), and an 1800; these stations treat the ESC 4 sequence (punch selection code) as data.

X=09 TRANSCODE Printer Message:

STX ESC / A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 ETX

This message is coded in TRANSCODE and is used to test the IBM 2780 printer. It is valid only for a 2780 on a switched line or a non-switched multipoint line.

X=10 TRANSCODE Punch Message:

STX ESC 4 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 ETX

This message is coded in TRANSCODE and is used to test the IBM 2780 card punch. It is valid only for a 2780 on a switched line or a non-switched multipoint line.

----

[1]Note: The text includes several terminal control characters, such as CR, HT, VT, and FF, that, when sent to an output device, cause the associated function to occur, if the device is capable of performing that function. For example, the HT or FF characters in text sent to a terminal printer will cause the printer to execute the horizontal tab and forms feed operations, if the printer is so equipped.

X=11 TRANSCODE Message:

STX SYN SYN A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 ETX

This message is coded in TRANSCODE and may be used to test either the card punch or the printer of an IBM 2780. It is valid only for a 2780, on any type of line configuration.

X=12 EBCDIC Printer Message:

This message has the same content as the TRANSCODE printer message, X=09, except coded in EBCDIC. This message is used to test the IBM 2780 printer. It may also be sent to a S/360 (including Model 20), 1130, 1800, and 2972; these stations treat the ESC / sequence (printer selection code) as data.

X=13 EBCDIC Punch Message:

This message has the same content as the TRANSCODE punch message, X=10, except coded in EBCDIC. This message is used to test the IBM 2780 card punch. It may also be sent to a S/360 (including Model 20), 1130, 1800, and 2972; these stations treat the ESC 4 sequence (punch selection code) as data.

X=14 EBCDIC Alphameric Message:

This message has the same content as the TRANSCODE message, X=11, except coded in EBCDIC. This message may be used to test either the card punch or the printer of an IBM 2780. It may also be sent to a S/360 (including Model 20), System/3, 1130, 1800, and 2770.

X=15 EBCDIC Weak Pattern Message[1]

STX SYN SYN ...text... ETX
The text consists of 74 NUL (X'00') characters, followed by six SYN (X'32') characters. This message may be sent to a S/360 (including Model 20), System/3, 1130, 1800, 2770, and 2780.

X=16 EBCDIC Weak Pattern Message[2]

STX SYN SYN ...text... ETX
The text consists of 40 bytes of X'AA', followed by 40 bytes of X'55'. This message may be sent to a S/360 (including Model 20), System/3, 1130, 1800, 2770, and 2780.

X=17 Transcode Weak Pattern Message[1]

STX SYN SYN ...text... ETX
The text consists of 80 SOH (X'00') characters. This message may be sent only to a 2780.

X=18 Transcode Weak Pattern Message[2]

STX SYN SYN ...text... ETX
The text consists of 40 N's (X'15'), followed by 40 ESC (X'2A') characters. This message may be sent only to a 2780.

X=19 EBCDIC Weak Pattern Message (DLE SYN Insertion)

DLE STX ...text... DLE ETX
The text consists of 280 NUL (X'00') characters, followed by 10 SYN (X'32') characters. This message may be sent to a S/360 (including Model 20), System/3, 1130, 1800, and 2715.

X=20 Transparent EBCDIC Message

DLE STX ...text... DLE ETX
The text consists of the characters U through Z, 0 through 9, and X'00' through X'3F' (a total of 80 characters). This message may be sent to a S/360 (excluding Model 20), 1800, 2770, and 2780.

X=21 Transparent EBCDIC Message

DLE STX ...text... DLE ETX
The text consists of the characters A through Z, 0 through 9, and X'00' through X'53' (a total of 120 characters). This message may be sent to a S/360 (excluding Model 20), 1800, 2770 and 2780.

X=22 Transparent EBCDIC Message

DLE STX ...text... DLE ETX
The text consists of the characters A through Z, 0 through 9, and X'00' through X'6B' (a total of 144 characters). This message may be

----

[1]This test type is intended for use by the IBM Customer Engineer to test for proper functioning of the data set clock (for switched lines) or business machine clock (for switched or nonswitched lines).

[2]This test type is intended for use by the IBM Customer Engineer to test for proper functioning of the data set clock (for nonswitched lines).

sent to a S/360 (excluding Model 20), 1800, 2770, and 2780.

The two SYN characters following the STX in nontransparent test messages are present to allow space for a component selection address, if required in a point-to-point configuration. If a component selection address is not required in the message, the SYNs will be transmitted, but will be deleted by the receiving station.

X=23 3270 Basic Test Message (EBCDIC)

This test message checks all alphameric characters at a display station or printer. It checks the use of the WCC to sound the audible alarm and allows attribute field specification to be checked at a display station. It starts a printer, printing 40 characters to a line.

X=24 3270 Model 1 Align Test Pattern (EBCDIC)

This test pattern checks position alignment for the 480-character display station. It also checks the WCC for sounding the audible alarm. It starts a printer, printing 40 characters to a line.

X=25 3270 Model 2 Align Test Pattern (EBCDIC)

This test pattern checks position alignment for the 1920-character display station. It also checks the WCC for sounding the audible alarm. It starts a printer, printing 80 characters to a line.

X=26 3270 Orders Test Message (EBCDIC)

This test message checks 3270 orders (for example, SF and SBA), checks the WCC for sounding the audible alarm, and uses high and normal intensities. It starts a printer, printing 64 characters to a line.

X=27 3270 Basic Printer Test Pattern (EBCDIC)

This test pattern, which is mainly intended for the printer, checks several solid lines of alphameric print. It checks the WCC for starting the printer and prints 132 characters to a line (honoring NL and EOM orders). (If issued to a display station, it checks the WCC for sounding the audible alarm.)

X=28 3270 NL/EOM Printer Test Pattern (EBCDIC)

This test pattern, which is mainly intended for the printer, checks the end of message (EOM) order and multiple new-line (NL) orders. It checks the WCC for starting the printer and prints 132 characters to a line. (If issued to a display station, it checks the WCC for sounding the audible alarm.)

X=29-34 3270 Test Messages and Patterns (ASCII)

These test messages and patterns correspond to types 23-28. ASCII transmission code is used instead of EBCDIC.

Table 47 shows the types of on-line tests that can be used for each type of remote station, except 3270 display stations and printers.

Table 47A shows the types of on-line tests that can be used for each remote 3270 display station or printer.

Console Messages

The on-line test facility prints on the console typewriter of the central computer the results of an on-line test. Messages are in one of two formats: For messages reporting the results of BTAM-transmitted test messages, or of a BTAM-transmitted RFT message specifying a test type (X field) of 00:

```
IEC807I cuu ON-LINE TEST xx yy tt nn
        ii...ii
```

For messages reporting the results of test messages received by BTAM from a remote computer or terminal:

```
IEC808I cuu ON-LINE TEST xx yy tt ll dd
```

The meanings of the message fields are:

cuu
: indicates the address of the line (channel and unit)

xx
: indicates the test type (X field of the RFT message).

The table shows the test types available for each type of remote station, and whether the RFT message that initiates the test can be sent from the central computer (indicated by "C") or from the remote station (indicated by "R"), or from both. Where a - appears, the test type is not usable for that type of remote station.

Test type - specified in RFT message X - field	Content of test message	S/360 except Mod 20	S/360 Mod 20	S/3	1130	1800	2715	2770	2780	2972
00	(User-specified)	C,R	C,R	C,R	C,R	C,R	R	C,R[1]	C,R[1]	C,R[1]
01	(User-specified)	C,R	R	R	R	R	R	R	R	R
02	EBCDIC, all bit patterns, transparent	C,R	R	R	R	R	R	R[3]	-	-
03	USASCII, all bit patterns, transparent	C,R	R	-	-	-	-	-	-	-
04	EBCDIC, all bit patterns except data link controls	C,R	R	R	R	R	-	R[3]	-	-
05	USASCII, all bit patterns except data link controls	C,R	R	R	-	R	-	R	R[2]	-
06	USASCII, A-Z, 0-9	C,R	R	R	-	R	-	R	R	-
07	USASCII, printer selection code and A-Z, 0-9	C,R[4]	R[4]	-	-	R[4]	-	R[4]	R[5]	-
08	USASCII, punch selection code and A-Z, 0-9	C,R[4]	R[4]	-	-	R[4]	-	R[4]	R[5]	-
09	Transcode, printer selection code and A-Z, 0-9	-	-	-	-	-	-	-	R[5]	-
10	Transcode, punch selection code and A-Z, 0-9	-	-	-	-	-	-	-	R[5]	-
11	Transcode, A-Z, 0-9	-	-	-	-	-	-	-	R	-
12	EBCDIC, printer selection code and A-Z, 0-9	C,R[4]	R[4]	-	R[4]	R[4]	-	R[4]	R[5]	-
13	EBCDIC, punch selection code and A-Z, 0-9	C,R[4]	R[4]	-	R[4]	R[4]	-	R[4]	R[5]	-
14	EBCDIC, A-Z, 0-9	C,R	R	R	R	R	-	R	R	-
15	EBCDIC, 74 NUL (X'00') characters 6 SYN (X '32') characters	C,R	R	R	R	R	-	R	R	-
16	EBCDIC, 40 bytes of X 'AA', 40 bytes of X '55'	C,R	R	R	R	R	-	R	R	-
17	Transcode, 80 SOH (X '00') characters	-	-	-	-	-	-	-	R	-
18	Transcode, 40 N's (X '15'), 40 ESC (X '2A') characters	-	-	-	-	-	-	-	R	-
19	EBCDIC, 280 NUL (X '00') characters 10 SYN (X '32') characters (transparent)	C,R	R	R	R	R	R	-	-	-
20	EBCDIC: 80 characters, U-Z, 0-9, X '00' - X '3F' (transparent)	C,R	-	-	-	R	-	R	R	-
21	EBCDIC, 120 characters: A-Z, 0-9, X '00' - X '53' (transparent)	C,R	-	-	-	R	-	R	R	-
22	EBCDIC, 144 characters: A-Z, 0-9, X '00' - X '6B' (transparent)	C,R	-	-	-	R	-	R	R	-

[1] RFT message sent from a remote 2770, 2780, or 2972 that specify X=00 must specify a transmission count (Y-field) of 1.

[2] Printer only. The print chain must be at least 120 characters.

[3] 2770s with expanded buffer capability.

[4] The printer and punch codes apply to the 2780 only; the devices that contain programming (S/360 Model 20, 1130, 1800) treat the codes as data.

[5] Nonswitched or switched point-to-point line only; not valid for multipoint.

Table 47. Summary of BSC On-Line Test Options (except for remote 3270 test options)

yy
: indicates the number of transmissions. For IEC807I messages, this value is obtained from the N field of the RFT message. For IEC808I messages, this value is accumulated by the on-line test routine as each test message is received by BTAM.

tt
: indicates the number of occurrences of timeout errors.

nn
: is the number of NAK responses to BTAM-transmitted test messages.

ii
: is the terminal identification sequence. This is printed for tests on multipoint lines.

ll
: indicates the number of occurrences of lost-data errors.

dd
: indicates the number of occurrences of data check errors.

ON-LINE TESTING FOR LOCAL 3270 DISPLAY SYSTEM

On-line tests for local 3270 devices are requested from local display stations. There are six types of tests, not all of which apply to all devices.

On-line testing between a System/360 computer and a local 3270 device requires that BTAM or an on-line diagnostic program be running in the computer. Only the local 3270 display system can initiate an on-line test. Tests are requested by sending a request-for-test (RFT) message to the computer. For a local 3270, (1) the cursor is positioned at the top left of an unfromatted screen (by pressing the CLEAR key and then the RESET key, for example), (2) the text of the RFT message (test type, number of times, length of address, channel and unit address) is entered by means of the keyboard, and (3) the TEST REQUEST key is pressed to precede the text with control characters (SOH % / STX).

TYPES OF TESTS

Types 23-28

The local 3270 display station sends an RFT message, which neither contains nor is followed by a test message. The X field of the RFT message indicates which of six BTAM-defined standard test messages it is to return to a local 3270 device. The Y field indicates how many times the test message is to be sent. The ADDR field indicates which local 3270 device is to receive the test message. The device receiving the test message must be associated with the same DCB as the device sending the RFT message. The contents of test messages are given below.

BTAM RESPONSE TO REQUEST-FOR-TEST MESSAGES

BTAM recognizes and responds to any RFT message received from a local 3270 display station provided that:

1. The on-line test facility is available (that is, T was specified among the EROPT options of the DCB macro instruction for the group of local 3270 devices).

2. The RFT message was received on a read initial operation (that is, a READ TI macro instruction was issued).

3. The input area is at least 300 bytes long (that is, the inlength operand of the READ macro instruction was at least 300). If buffering is used, the entire 300-byte area must be contained within one buffer.

4. The RFT message was received without error.

When BTAM recognizes an RFT message, control is given to the on-line test logic, which examines the message, generates the requested test message in the input area for the read operation, and sends the test message the requested number of times. If the test message was sent to the same device from which the RFT was received, the read initial operation is restarted. If the test message was sent to a different device, the read initial operation is posted complete with a completion code of X'44'.

FORMATS OF RFT, TEST, AND CONSOLE MESSAGES

Request-for-Test Message

An RFT message from a local 3270 display station has the format:

SOH %	/	STX	X	Y	N	ADDR
2	1	1	2	2	1	3

Field length (bytes)

SOH %
: identifies the message as an RFT message.

/
: identifies the message as an RFT message from a local 3270.

X
: specifies the test type (23-28). X is a two-byte zoned decimal field.

Y
: specifies the number of times (1-99) the test message is to be sent. Y is a two-byte zoned decimal field.

N
: specifies the length (3) of the ADDR field.

ADDR
: contains three characters (0-9, A-F) that indicate the channel and unit address of the device that is to receive the test message.

__Test Messages__

The contents of test messages are determined by the X (test type) field of the RFT message that initiates the sending of the test message.

__X=23-28 3270 Test Messages and Patterns (EBCDIC)__

These test messages and patterns correspond to types 23-28 for remote 3270 display stations and printers. See the descriptions of test messages 23-28 above under "On-line Testing for Binary Synchronous Communications Lines."

Table 47B shows the types of on-line tests that can be used for each local 3270 display station or printer.

__Console Messages__

See the description of console messages above under "On-Line Testing for Binary Synchronous Communications Lines."

X's indicate the test type available for each remote 3270 device.

		3271				3275		
		3277		3284 or 3286				
Test Type	Content of Test Message	Model 1	Model 2	Model 1	Model 2	Model 1	Model 2	3284 Model 3
23	3270 Basic (EBCDIC)	X	X	X	X	X	X	
24	3270 Model 1 (EBCDIC)	X		X		X		
25	3270 Model 2 (EBCDIC)		X		X		X	
26	3270 Orders (EBCDIC)	X	X	X	X	X	X	
27	3270 Basic Printer (EBCDIC)			X	X			X
28	3270 NL/EOM Printer (EBCDIC)			X	X			X
29	3270 Basic (ASCII)	X	X	X	X	X	X	
30	3270 Model 1 (ASCII)	X		X		X		
31	3270 Model 2 (ASCII)		X		X		X	
32	3270 Orders (ASCII)	X	X	X	X	X	X	
33	3270 Basic Printer (ASCII)			X	X			X
34	3270 NL/EOM Printer (ASCII)			X	X			X

Table 47A. Summary of BSC On-Line Test Options for Remote 3270 Devices

X's indicate the test types available for each local 3270 device.					
Test Type	Content of Test Message	3272			
		3277		3284 or 3286	
		Model 1	Model 2	Model 1	Model 2
23	3270 Basic	X	X	X	X
24	3270 Model 1	X		X	
25	3270 Model 2		X		X
26	3270 Orders	X	X	X	X
27	3270 Basic Printer			X	X
28	3270 NL/EOM Printer			X	X

Table 47B. Summary of On-Line Test Options for Local 3270 Devices

This appendix illustrates each of the various kinds of terminal lists given under Defining and Modifying Terminal Lists, elsewhere in this manual.

Each terminal list consists of one or more entries, each representing a remote station or a specific component of a remote station. Terminal lists vary in format; the illustrations in this appendix show how each type is organized.

Note: Terminal lists are not used for the local 3270 display system.

In lists of the OPENLST and WRAPLST type, each entry contains a control byte, illustrated in Figure 31. The bits in this control byte have the following meaning:

Bit Position	Meaning
0	If on, indicates that the entry is the last in the list.
1	If on, indicates that the entry is to be skipped when polling or addressing. If off, indicates an active entry. This bit is turned on and off with the CHGNTRY macro.
2	If on, indicates that the list is a wraparound list.
3-7	List entry number. Each entry is numbered successively starting with 1. This field limits to 31 the number of terminal or component entries for a list created by the DFTRMLST macro. This field is not presently used by BTAM, but is reserved for later use. Large lists can be created by coding a series of DFTRMLST macro instructions of the OPENLST type. If a wraparound list is desired, code a series of DFTRMLST macros of the OPENLST type, and follow the last in the series by the instruction DC HL2'-n', where n is the number of bytes occupied by the terminal list entries.

Note: In the examples the polling and addressing characters and the identification sequences are shown as alphabetic and numeric characters, but you must code them in the DFTRMLST macro as the hexadecimal representation of the appropriate transmission code bit patterns.

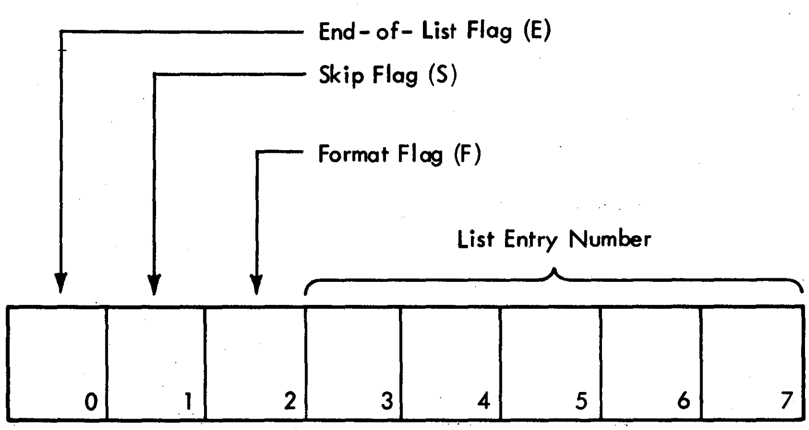
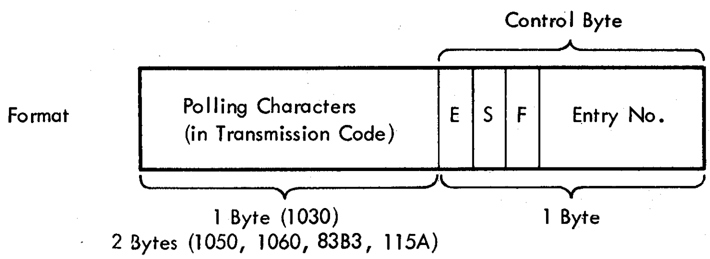


Figure 31. Format of Control Byte for OPENLST and WRAPLST Entries

OPENLST Format

Each entry in an open list (polling or addressing) consists of a one (1030) or two (all others) byte field for the polling or addressing characters plus the control byte. Examples for 1050 and 1030 are shown in Figure 32.



Example IBM 1050

A	5	0	0	0	0	0	0	0	1
A	6	0	1	0	0	0	0	1	0
B	5	0	0	0	0	0	0	1	1
B	6	0	0	0	0	0	1	0	0
C	5	0	1	0	0	0	1	0	1
E	0	1	0	0	0	0	1	1	0

Example IBM 1030

D	0	0	0	0	0	0	0	1
E	0	0	0	0	0	0	1	0
F	1	0	0	0	0	0	1	1

Figure 32. Open Polling or Addressing List (OPENLST): Format and Examples

WRAPLST Format

Wraparound polling lists differ from open lists in two ways:

1. Format bit (bit 2 in control byte) is on in the last entry.
2. A two-byte field follows the last entry and contains a negative binary value used by the polling restart routine to find the start of the list.

An example is shown in Figure 33.

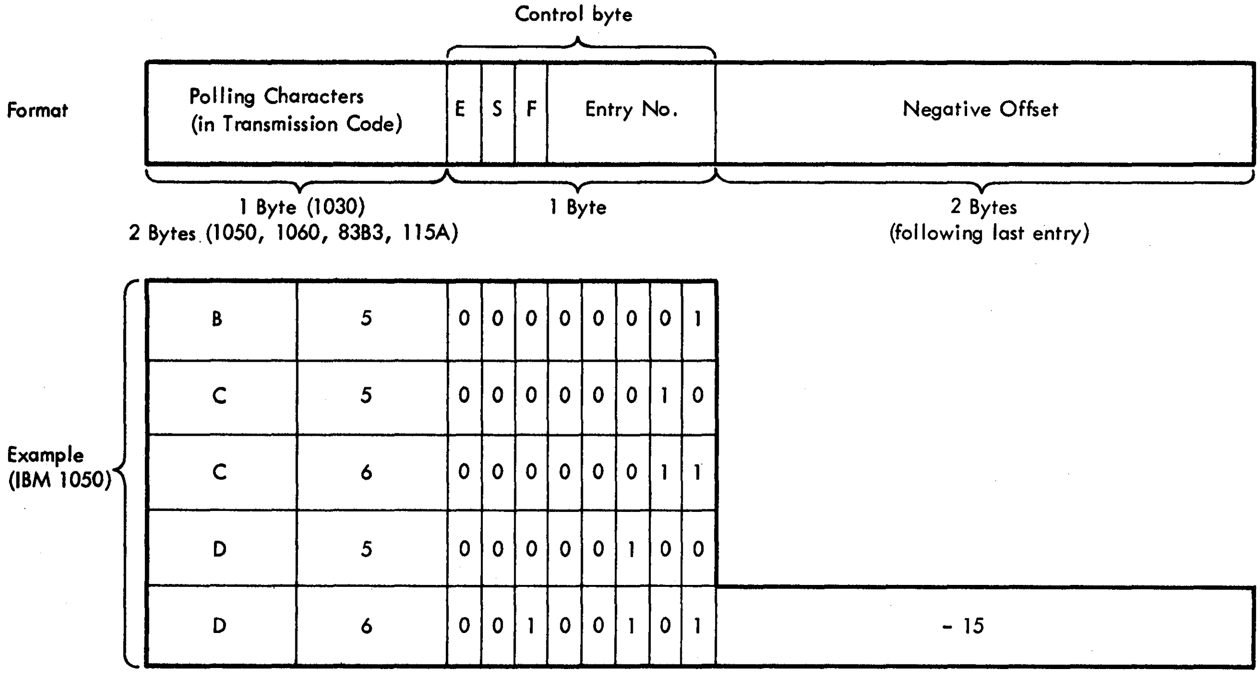


Figure 33. Wraparound Polling List (WRAPLST): Format and Example

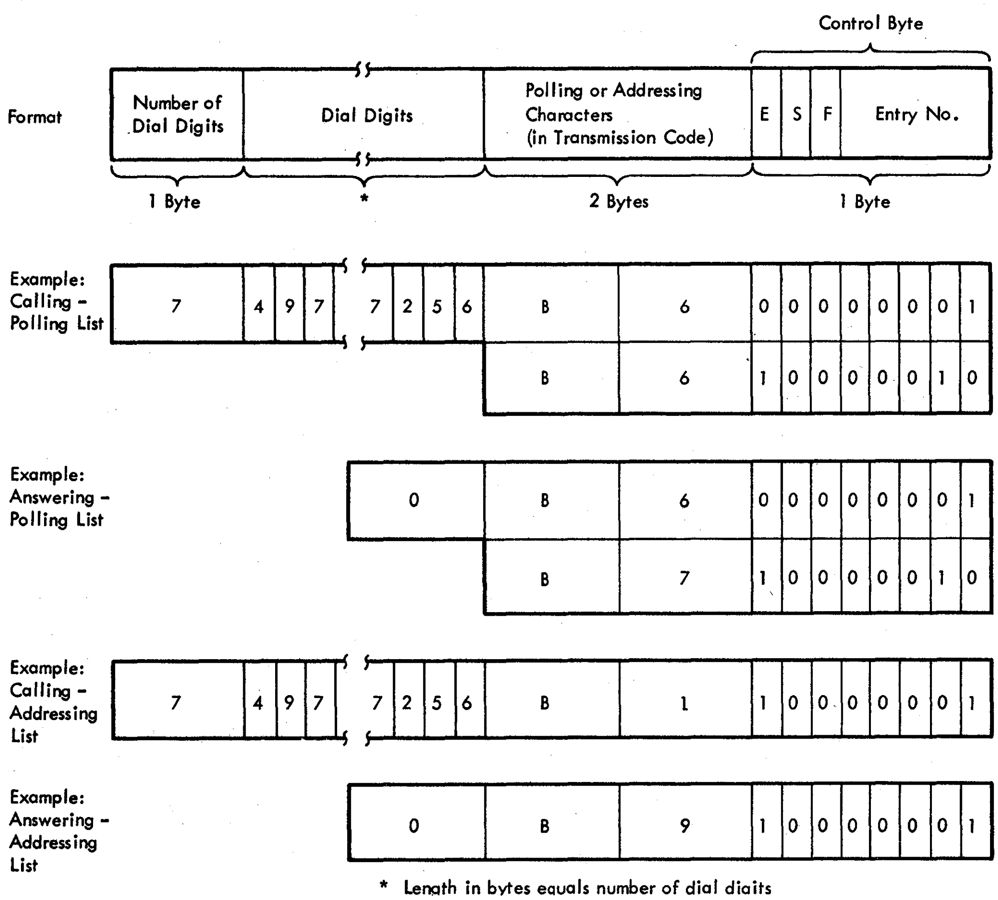


Figure 34. Dial List (DIALST): Format and Examples

DIALST Format

Terminal lists for stations on switched lines are illustrated in Figure 34.

IDLST Format

Terminal lists for TWX terminals (Models 33 or 35) are illustrated in Figure 35.

SSALST and SSAWLST, AUTOLST and AUTOWLST Format

Terminal lists for all stations for which Auto Poll is employed are illustrated in Figure 36.

TE
: the total number of entries in list (1-253)*

AE
: the total number of active entries in list (0-253)*

NNN
: entry width Pi+Ii (2-7)

W
: Wraparound flag (on for SSAWLST and AUTOWLST, off for SSALST and AUTOLST)

UC
: Usage Count (0-15). The usage count indicates the total number of polling operations using the terminal list at any one time.

Pi
: polling characters (1 or 2 bytes). The value X'FE' must not be used as a polling character.

Ii
: index (1-253)*

X'FE'
: Scan stop byte used to find end of list.

OFFSET
: 2-byte field used to find heading of list from end of list.

*
: TE, AE, and Ii can be as high as 253, but at the time of publication, the assembler imposes additional restrictions.

BSCLST Format

Terminal lists for S/360-to-S/360 communication over a switched line are illustrated in Figure 37.

WTTALST Format

Terminal lists for World Trade telegraph terminals are illustrated in Figure 38.

SWLST Format

The format and contents of the header and entries of a calling and answering list of the SWLST form is as follows (see Figure 39).

Field	Contents
(HEADER) Pointer to Sequence Matching Received Sequence:	Address (right-adjusted) of the last authorized ID sequence that was received prior to completion of the READ Connect or WRITE Connect operation. (Byte 0 contains X'FF' to indicate that the list is of the SWLST form.)
Number of List Entries:	Number (binary) of entries in the list (i.e., the number of different authorized ID sequences that will be honored).
Entry Length:	Number of bytes (binary) in each entry in the list. This number is specified by the <u>entry-length</u> operand of the DFTRMLST macro, and should equal the number of bytes required to accommodate the longest expected ID sequence, plus the <u>user-data</u> field (0 or 4), plus one (for the control byte).
Read-In Area Length:	Number (binary) of characters in the longest expected ID sequence. This number will have a minimum value of 2, to accommodate a two-character sequence such as DLE EOT.
Dial Count:	For an automatic dialing list: number (binary) of dial digits to be used in calling the remote station. For a manual dialing list or an answering list: 0.
Read-In Area:	Area into which the ID response is read from the remote station. The length of this field is determined by the longest possible sequence that can be received, but no less than 2 bytes.
Dial Digits:	The dial digits (binary), for an automatic dial calling list. For an answering list or a manual-dial calling list, this field is omitted.
Id Count:	Number of characters (binary) in the sequence defined in the Id Sent field.
Id Sent	For a calling list, this field contains the characters of the ID-ENQ sequence to be sent to the remote station. For an answering list, this field contains the ID ACK-0 sequence to be sent to the remote station when the control byte value of the entry containing the received ID ENQ sequence is 0. It is recommended that the first two characters of each ID sequence be identical, to provide greater identification reliability.
(ENTRY) Authorized Sequence:	The characters composing an authorized sequence that can be received. The size of this field is usually the length of the Read-In Area. Since this length is never less than 2, the size of

this field is less than the size of the Read-In Area when an answering list is defined with only one entry, containing the single ENQ character. Authorized sequences can be of different lengths; each sequence is left-adjusted in the Authorized Sequence field.

User Area (optional): May contain a user-specified relocatable expression for each list entry. This four-byte field is included in each entry if you specify the <u>userlength</u> operand of the DFTRMLST macro as 4; otherwise, the field is omitted.

Control Byte: A value, specified in the <u>controlvalue</u> operand of the DFTRMLST macro, indicating the action BTAM is to perform when an authorized ID sequence is received. The value may be 0, 1, or 2. (See description of the DFTRMLST macro for the significance of these values.)

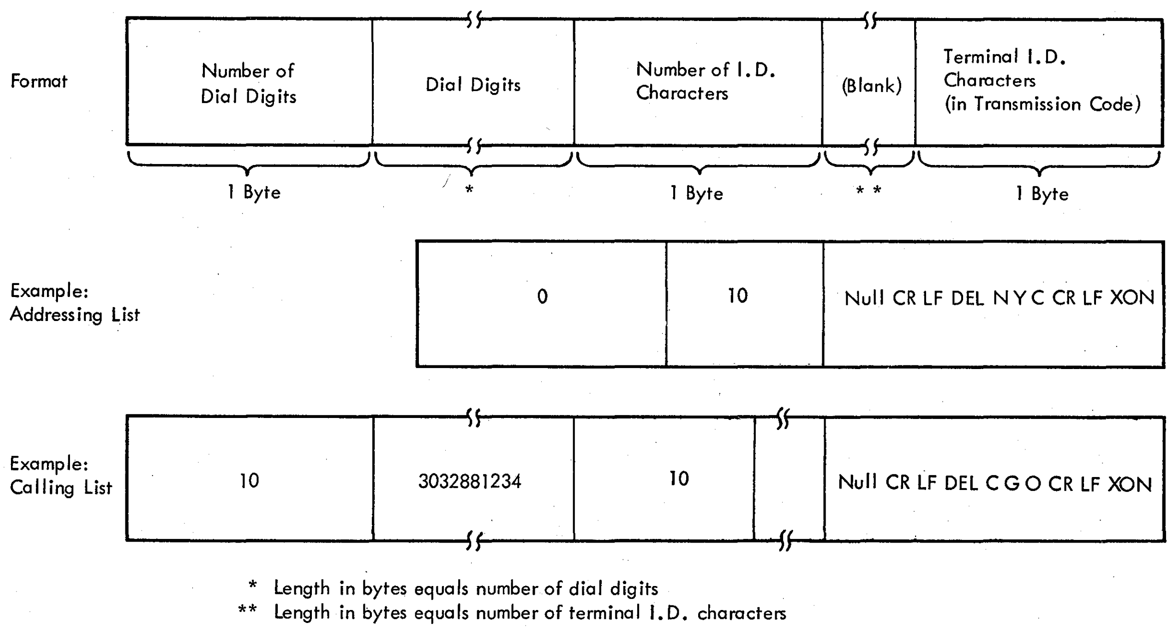


Figure 35. Identification List (IDLST): Format and Example

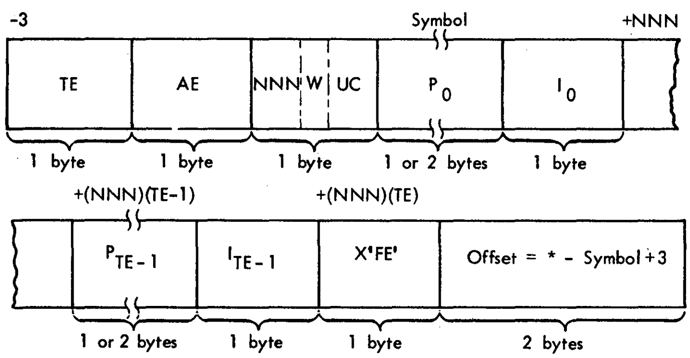
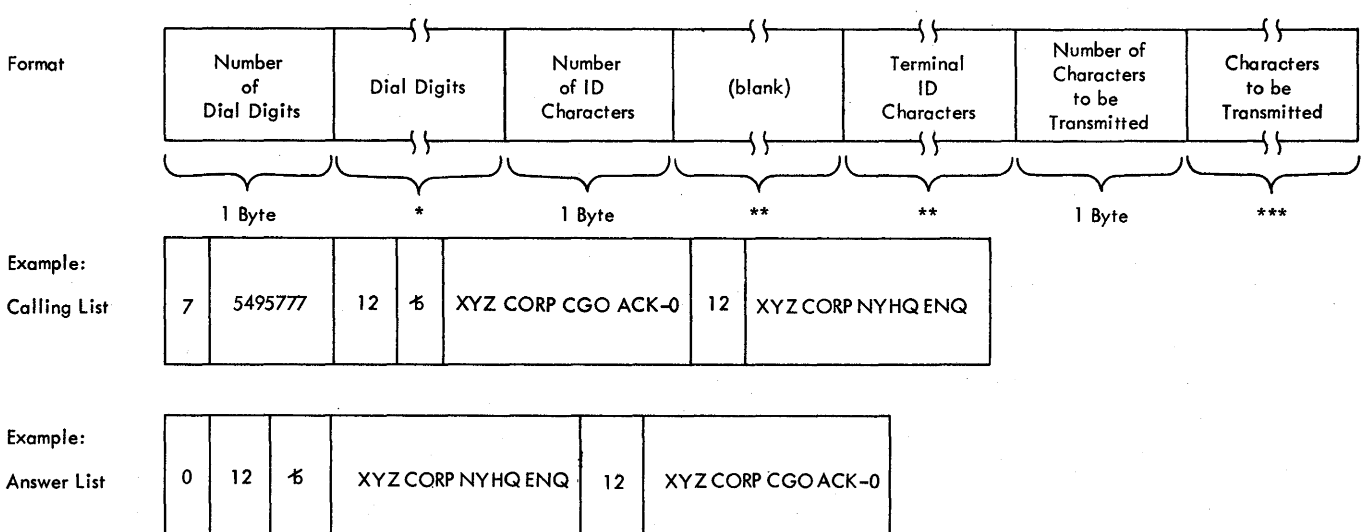


Figure 36. Open and Wraparound Auto Poll Lists for Start-Stop (SSALST, SSAWLST) and BSC (AUTOLST, AUTOWLST): Format

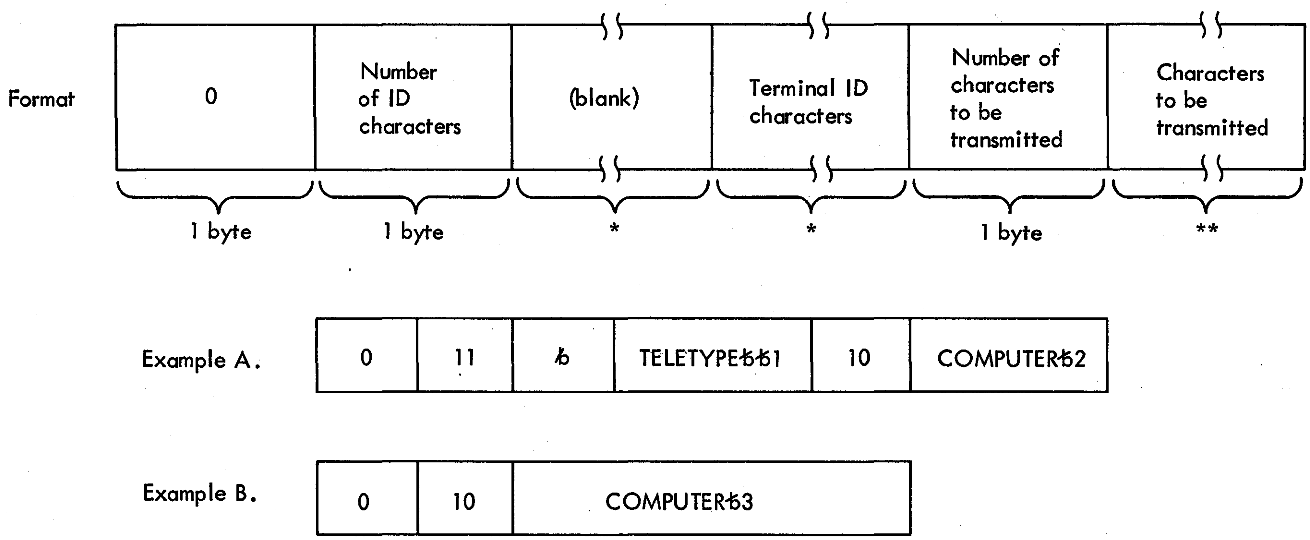


The control characters are coded in their hexadecimal equivalents shown below:

ENQ - X'2D'
ACK-0 - X'1070'

* Length in bytes equals number of dial digits.
** Length in bytes equals number of terminal ID characters.
*** Length in bytes equals number of characters to be transmitted.

Figure 37. BSC Dial List (BSCLST) (for S/360-S/360): Format and Example



If the transmission code used with the WT terminals is the International Telegraph Alphabet No. 2, these terminal lists would be defined by coding:

<u>Example A</u>:

```
DFTRMLST   WTTALST,0,11,0110091001150D1004043D,10,0E03070D1C01100A0439
```

<u>Example B</u>:

```
DFTRMLST   WTTALST,0,0,0,10,0E03070D1C01100A0430
```

*length in bytes equals the number of terminal ID characters to be received.

**Length in bytes equals number of computer ID characters to be transmitted.

Figure 38. WT Terminal List (WTTALST): Format and Examples

General Format of List:

Header	Entry 1	Entry 2	. . .	Entry n

Header of Auto-Dial Calling List:

	Pointer to Sequence Matching Received Sequence	Number of List Entries	Entry Length	Read-in Area Length	Dial Count	Read-In Area	Dial Digits	ID Count	ID Sent
Length in bytes:	4	1	1	1	1	=Read-In Area Length	=Dial Count	1	=ID Count

Header of Manual-Dial Calling List or Answering List:

	Pointer to Sequence Matching Received Sequence	Number of List Entries	Entry Length	Read-In Area Length	0	Read-In Area	ID Count	ID Sent
Length in bytes:	4	1	1	1	1	=Read-In Area Length	1	=ID Count

Entry including User Area:

	Authorized Sequence	User Area	Control Byte
Length in bytes:	=Read-In Area Length	4	1

Entry omitting User Area:

	Authorized Sequence	Control Byte
Length in bytes:	=Read-In Area Length	1

Figure 39. Calling and Answering Lists for Expanded ID Verification (SWLST): Format

DECSDECB

Standard 4-byte ECB. Only the first byte of this field is of concern to the BTAM programmer. This byte can contain the following hexadecimal completion codes:

Hex Value	Meaning
00	After Read or Write macro instruction issued, before WAIT.
80	WAIT macro instruction issued; event not complete. Note: As long as the wait bit is on, the contents of some DECB fields are unpredictable (the fields are used internally by BTAM); the contents of the DECB should therefore be considered meaningful only after the Read or Write operation has been completed (i.e., the completion bit is on).
7F	Normal completion: The Read or Write operation has ended with indications of Channel End-Device End and either Unit Exception or Incorrect Length, or both, if they are normal conditions (e.g., Unit Exception indicating end-of-transmission or negative response to polling). The user program should examine the bits in DECFLAGS to determine the status of the operation.
41	Complete with I/O error; the program should examine the bits in DECERRST to determine the kind of error.
44	The I/O request was rejected, because (1) a device error was detected after the last I/O operation on the device was posted complete or (2) a request-for-test message was received from a local 3270 display station requesting that a test message be sent to another local 3270 device.
48	Enable Command Halted or I/O Operation Purged: Indicates one of the following: • An Enable command (automatic answering function for a switched line) was terminated by Halt I/O as a result of a RESETPL macro instruction (second operand omitted or specified as ANSRING). • An Enable command was terminated as a result of closing (CLOSE macro instruction) a line group with Enable commands outstanding. • An I/O operation was purged at Channel End interrupt time as a result of closing the line group while I/O operations were still in progress. • A Read Initial operation for World Trade telegraph has ended with a Halt I/O command because a RESETPL macro instruction was issued (second operand omitted).

- **A Read Initial operation for the local 3270 display system was canceled, because a RESETPL macro instruction was issued.**

DECTYPE Operation type:

first byte: (In any combination)

bit 0 - current operation is a Read operation using Auto Poll

Exception: When BSC on-line test is in control of the line, the 0 and 1 bits have the following meaning:

bit 0 - indicates that on-line test has been requested by the ONLTST macro

bit 1 - (meaningful only when on-line test has been initiated) =0 if test messages are sent by BTAM; =1 if test messages are received by BTAM =1 if RJE (Remote Job Entry) requested WTO timeout message suppression

bits 2-4 - (reserved)
bit 5 - 'entry' coded as 'S'
bit 6 - 'area' coded as 'S'
bit 7 - 'length' coded as 'S'

second byte: bits 3, 4, 5, 6, and 7

Hex Value	Operation
00	Write Break (TB)
01	Read Initial (TI)
02	Write Initial (TI)
03	Read Continue (TT)
04	Write Continue (TT)
05	Read Conversational (TV)
06	Write Conversational (TV)
07	Read Repeat (TP), or Read Continue with Identification Exchange (TE) (WT terminal)
08	Write Positive Acknowledgment (TA)
09	Read Skip (TS)
0A	Write Negative Acknowledgment (TN), Write Reset (TR), Write Disconnect (TN) (TWX)
0B	Read Buffer (TB)
0C	Write at Line Address (TL), Write Initial Optical (TIO), Write Initial Transparent Block (TIE)
0D	Write Initial Conversational (TIV), Read Continue with Leading Acknowledgment (TTA)
0E	Write Erase (TS), Write Invitational Optical (TCO), Write Continue Transparent Block (TTE)
0F	Write Continue Conversational (TTV)
10	Write Disconnect (TD) (BSC)
11	Read Connect (TC), Read Modified (TM)
12	Write Initial Transparent (TIX), Write Conversational Optical (TVO), Write Unprotested Erase (TUS)
13	Read Continue with Leading Graphics (TTL), Read Buffer from Position (TBP)
14	Write Continue Transparent (TTX)
15	Read Inquiry (TQ)
16	Write Inquiry (TQ)
17	Read Repeat with Leading Graphics (TPL)
18	(Reserved)
19	Read Initial Inquiry (TIQ), Read Modified from Position (TMP)
1A	Write Wait Before Transmitting (TW)
1B	Read Interrupt (TRV)
1C	Write Connect (TC)
1D	Write Initial Conversational Transparent (TIVX)
1E	Read Connect with Tone (TCW)
1F	Write Continue Conversational Transparent (TTVX)

Bit 0 of this second byte specifies Reset for Read Initial and Reset (TIR), Write Initial and Reset (TIR), Read Continue and Reset (TTR), Write Continue and Reset (TTR), Read Conversational and Reset (TVR), Write Conversational and Reset (TVR), Read Repeat and Reset (TPR), Read Buffer and Reset (TBR), Write at line Address and Reset (TLR), and Write Erase and Reset (TSR). Bits 1 and 2 are reserved.

DECLNGTH
Buffer length or message area length.

DECONLTT
(Reserved)

DECDCBAD
Address of associated DCB.

DECAREA
Address of the message area or first buffer. The high-order byte of this field must always contain zero.

DECSENS0
Sense information, as set by the control unit, when the CSW status (DECCSWST) indicates a unit check.

Bit	Meaning
0	Command reject
1	Intervention required
2	Bus out check
3	Equipment check
4	Data check
5	Overrun
6	Lost data
7	Timeout

DECSENS1
(Reserved)

DECCOUNT
Residual count from the CSW for the last CCW that was executed.

DECCMCOD
Command Code (one byte) identifies the type of command upon which the error occurred.

Hex Value	Command
01	Write
02	Read
03	I/O No-op
04	Sense
06	Prepare
09	Poll
0A	Inhibit
0D	Break
27	Enable
29	Dial
2F	Disable

DECENTRY
Address of the terminal list entry specified in the entry operand of the READ or WRITE macro instruction, prior to a Read or Write operation; after the operation it contains the next sequential address of the terminal list if program polling and a terminal list of the WRAPLST type are specified.

DECFLAGS
Status flags that may be set regardless of whether there was an I/O error (i.e., the completion code in the DECSDECB may be either 7F or 41).

Bit 0: For start-stop operations, this bit is reserved. For BSC operations, it indicates that a WACK (Wait-before-transmit) was received, if bit 1 is also on. If bit 1 is not on, bit 0 indicates that an error status message was received. (An error status message begins with SOH % S and provides status information about a remote station.) If a WACK has been received, the user program should respond by sending ENQ (or EOT, if transmission is to be ended), unless the WACK was received in response to selection (multipoint line), in which case the user program should retransmit the selection characters, that is, reissue the WRITE macro.

Bit 1: For start-stop operations, this bit is reserved. For BSC operations, it indicates that some response other than ACK-0 or ACK-1 was received into the DECRESPN field. Examination of the response will determine which action should be taken to reestablish proper communication. (This bit is set when WACK (see also bit 0) is received or when RVI (see also bit 6) is received.

Bit 2: For start-stop operations, this bit is reserved. For BSC operations, it means that an incorrect alternating acknowledgment was received: ACK-1 received when ACK-0 was expected, or vice versa. If this bit is on <u>and</u> the completion code for the operation is 7F (i.e., no line transmission error occurred), a complete message may have been lost.

Bit 3: The ID received from a TWX 33/35 or a BSC station did not equal the expected ID as defined in the terminal list specified in the WRITE TI, WRITE TC, or READ TC macro instruction, or the index received as a result of an Auto Poll operation did not match the index byte in any of the active entries in the polling list. For BSC (non-switched line) this bit, when on, indicates that contention has occurred and this is not the control station. The control station should retry this WRITE and this (remote) station should issue a READ Initial. For World Trade telegraph terminals, this bit indicates that contention occurred, or that the ID received from a terminal did not equal the expected ID as defined in the terminal list specified in the READ TE macro instruction. Test the TP code in the DECB to determine which condition occurred.

Bit 4: No buffer was available upon completion of a dynamic buffering Read command. The last buffer is posted complete and the remainder of the message is read from the communications line (under control of a dynamic buffering Read Skip command), but the data is not placed into storage.

Bit 5:

- The end of the terminal list has been reached, or all the skip bits are on. This is an indication that:

  1. A negative response to polling has been received from the terminal represented by the last active (non-skipped) entry in an open polling list (OPENLST, SSALST, AUTOLST);
  2. A negative response to polling has been received following a RESETPL macro instruction of the POLLING type (second operand omitted or specified as POLLING);
  3. All of the entries in a wraparound polling list (WRAPLST) are inactive (all skip bits are on).

  <u>Note</u>: Condition 3 can occur only as a result of one or more skip bits being turned on after initiation of a programmed polling operation with a wraparound polling list. If all skip bits were on at the time that the READ macro instruction was executed, no I/O operation would be initiated.

- Negative response to addressing has been received.

- The last message sent by a World Trade telegraph terminal ended with EOT or a time-out.

- For 2741: Power is off or other Intervention Required condition exists.

Bit 6: <u>WT Terminals:</u> Message ended with WRU signal.
<u>BSC Stations:</u> RVI sequence received (see also bit 1).
<u>2741</u>: Write operation was ended by terminal interrupt.

Remote 3270: If bit 6 is on, but bit 1 is not on, an error status message was received. (An error status message for a remote 3270 device begins with SOH % R and provides sense and status information about the device.)

Bit 7: WT Terminals: Contention condition was encountered.
BSC Stations: STX ENQ sequence was received.

Local 3270: OLTEP is using the device to run diagnostics.

DECRLN
Relative line number.

DECRESPN
Start-stop: First byte: one-character response to addressing
Second byte: one-character LRC/VRC response to text
BSC: two-character response to addressing, ENQ, or text. Exception: responses to text for Write TIV, TIVX, TTV, and TTVX are read into the input area designated by the WRITE macro.

DECTPCOD
TP Op. code. Bits 2-7 of these codes identify types of channel commands that are not identifiable by the command code alone. Bits 0 and 1 are used in conjunction with, but independent of, bits 2-7, as described below.

Hex Value	Meanings
00	Any command issued by On-line Test routine.
01	Disable, when the disable is the first command of a channel program; dial, enable, prepare, write pad characters, or write wait-before-transmitting; or sense (World Trade telegraph terminals).
02	Write EOA EOT EOT EOT sequence prior to selection, write EOT sequence prior to polling or addressing, write response to text, write EOA and 15 idle characters (Basic 2740), or Write EOA PRE o (2740/2760).
03	Write polling or addressing character or write / (/ is the broadcast addressing character) (2740 with Station Control), turn-around sequence (TWX), CPU-ID sequence (TWX or BSC), Poll command with SSALST, SSAWLST, AUTOLST, or AUTOWLST, or write inquiry (ENQ).
04	Write space (2740 with Station Control), write 2848 command (2260R), write FIGS (83B3), write 1 (1030), write WRU, Identification, pad, or LTRS characters (World Trade terminals), or Sense (2740).
05	Read response to polling.
06	Read response to addressing.
07	Read ID response (TWX or BSC).

08 Write end of addressing character following addressing (on 1030, 1050, 1060, 2260R, or 2740). Write response to inquiry. Write response to text (BSC). Write EOB (2760/2740).

09 NOP or TIC following Poll in the polling list: SSALST, SSAWLST, AUTOLST, or AUTOWLST.

0A Read index (Auto Poll) or read response to polling (programmed polling).

0B Read inquiry (BSC only).

0C Read response to inquiry (BSC only).

10 Write at line address (2260R).

11 Read or write text. Write frame change sequence (2760/2740).

12 Read skip or TIC command for dynamic buffering.

13 Write end-of-transparent text (DLE ETX) characters (BSC).

14 (Reserved)

20 Read response to text (start-stop).

21 All reset commands.

22 Read skip.

23 Write break.

24 Any command issued during OPEN, LOPEN, or CLOSE (Set Address, Enable, Disable, and Set Mode commands).

25 Read Response to text (BSC).

Bit 0: Indicates the final command in the channel program (not necessarily the last command executed).

Bit 1: The command just executed was the first Read Text or Write Text CCW to be executed in a channel program using dynamic buffering.

DECERRST

Error status flags that may be set if an I/O error has occurred (i.e., a completion code of 41 is placed in DECSDECB).

Bit 0: The START I/O instruction resulted in a condition code of 3, indicating that the control unit or the specified line is not operational.

Bit 1: An error condition that should not occur (is undefined for the particular command or device) has occurred.

Bit 2: An error condition occurred on an I/O operation initiated by the error recovery routines: (1) as part of an intermediate recovery procedure, (2) as part of a diagnostic write/read procedure (2701 only), or (3) as part of a disconnect procedure for a switched line.

Bit 3: A diagnostic write/read operation terminated in error, indicating a control unit failure (2701 only). An error occurred that makes the integrity of the device regeneration buffer doubtful (local 3270 only).

Bit 4: A Disable command was issued to a switched line by the error recovery routines after detecting a permanent error on that line.

Note: If this bit is on after execution of error recovery procedures, the user program must execute an initial-type Read or Write operation, in order to reestablish the line connection.

Bits 5-7: (Reserved)

DECCSWST
Contains the status bits from the CSW for the last CCW that was executed.

DECADRPT
Pointer to the addressing list entry used in the previous operation.

DECPOLPT
For programmed polling, contains the address of the current entry in the polling list. For Auto Poll, the high-order byte contains the index to the current polling list entry. The remaining bytes contain the address of the polling list (i.e., the address of the first entry therein). For BSC on-line test operations, contains the address of the area in which user-specified text data is placed (for test messages). For local 3270 read operations, contains the relative line number of the device from which the message was read.

DECWLNG
Length of the data area in leading-graphics or conversational operations or when using READ TWC.

DECWAREA
Address of the data area in leading-graphics and conversational operations or when using READ TWC. The high-order byte of this field must always contain zeros.

Fields Defined by User

It may be useful for the user program to maintain application-dependent information about the line and about the stations connected to the line. This may conveniently be done by appending to each DECB a sequence of fields containing the needed information, which might typically include:

- Line status: A one-byte field that indicates the status of the line; for example, active or inactive. The inactive bit might be set after a certain number of transmission errors have accumulated, to indicate to the message control routine that no further Read and Write operations are to be executed using that line.

- Address of User Terminal Table: This table would contain a series of fixed-length entries, one for each terminal, containing terminal information such as whether or not the terminal is active, and the addresses of the terminal list entries for that terminal.

- Terminal Count: A count of the number of terminals connected to the line.

- Processing Routine Address: Contains the address of the next routine to be given control for the line. For example, this field would contain the address of a line analysis routine to be given control upon completion of a Read or Write operation.

	Offset	Field	Contents
Standard DECB Fields	0	DECSDECB	Standard Event Control Block
	4	DECTYPE	Operation Type
		DECLNGTH	Message Area Length or Buffer Length
	8	DECONLTT / DECDCBAD	Reserved
			Address of DCB
	12	DECAREA	Reserved
			Address of Message Area or Address of First Buffer
DECB Fields Peculiar to BTAM	16	DECSENS0	Sense Byte
		DECSENS1	Reserved
		DECCOUNT	CSW Residual Count
	20	DECCMCOD / DECENTRY	Command Code
			Address of Terminal List or of an Entry Therein
	24	DECFLAGS	Status Flags
		DECRLN	Relative Line Number
		DECRESPN	Response to Addressing (1 byte for Start/Stop, 2 bytes for BSC)
			Response to VRC/LRC (Start/Stop Only)
	28	DECTPCOD	TP-OP Code
		DECERRST	I/O Error Status
		DECCSWST	CSW Status
	32	DECADRPT	Reserved
			Pointer to Previous (or Only) Entry in Addressing List
	36	DECPOLPT	Index byte (Auto Poll only) Relative line number (local 3270 only)
			Address of Current Entry in Polling List (Programmed Polling) Address of Polling List (Auto Poll)
Present only in DECBs for BSC lines	40		Reserved
		DECWLNG	Length of Data Area (for BSC Write Conversational and BSC Read with Leading Graphics operations)
	44	DECWAREA	Reserved
			Address of Data Area (for BSC Write Conversational and BSC Read with Leading Graphics operations)
Fields Defined By User			Line Status
			Address of User Terminal Table
			Terminal Count
			Polling List Address Processing Routine Address
			Other User - Defined Data

Figure 40. Format of Data Event Control Block

This appendix explains each of the BTAM-related error messages that may be printed during program execution at the console of the central computer, or at some other console, if the system includes the Multiple Console Support facility, or in the assembler listing in the SYSPRINT data set during program assembly. Also given are Abend codes 090 - 098 which may be issued during opening of a BTAM DCB.

Both system-generated and user-generated messages are described herein. System-generated refers to those messages printed at a console (or in an assembly listing) by the operating system; these begin with a standard identification, e.g., IEC801I.

User-generated refers to those messages that are sent by the user (e.g., the operator of a remote station) to the central computer for routing to the user program, console, or an error file on a system residence device.

ASSEMBLY ERRORS

These messages are produced by the assembler program during expansion of supervisor and data management macro instructions. They appear on the assembler listing in the SYSPRINT data set.

IHB002 INVALID xxx OPERAND SPECIFIED-yyy

Explanation: An operand whose position or name is xxx was specified as yyy. The specified operand is invalid.

System Action: The macro instruction was partially expanded; expansion stopped on detection of the error. Severity code = 12.

Programmer Response: Probable user error. Correct the invalid operand and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB072 LERB REQUESTED - EROPT=C ASSUMED

Explanation: LERB was coded in the DCB but EROPT=C (indicating a request for line error recording) was not coded.

System Action: The macro instruction was expanded normally with line error recording provided. Severity code=*.

Programmer Response: Probable user error. Delete the LERB operand if line error recording is not wanted. If line error recording is wanted, code EROPT=C. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB073 LERB OMITTED - ERROPT=C IGNORED

Explanation: Line error recording was requested by EROPT=C but no LERB address was given.

System Action: The macro instruction was expanded normally with no line error recording provided. Severity code=*.

Programmer Response: Probable user error. Delete EROPT=C if line error recording is not wanted. If line error recording is wanted, code a LERB address. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB074 EROPT=N - LERB IGNORED

Explanation: LERB was coded in the DCB but error recovery procedures were not requested (EROPT=N).

System Action: The macro instruction was expanded normally with no line error recording provided. Severity code=*.

Programmer Response: Probable user error. Delete the LERB operand if line error recording is not wanted. If line error recording is wanted, code EROPT=C. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB075 TABLENAME OPERAND REPEATED - XXX

Explanation: In the ASMTRTAB macro instruction, a table name operand was coded more than once. XXX is the repeated operand.

System Action: The macro instruction was expanded normally. Severity code=*.

Programmer Response: Probable user error. Remove the duplicate operand and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB076 MACRO NAME FIELD BLANK - NAME REQUIRED

Explanation: A name must be specified in the name field for this macro instruction.

System Action: The macro instruction was not expanded. Severity code=12.

Programmer Response: Probable user error. Code a name in the name field of the macro instruction and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB078 XXX OPERAND REGISTER NOTATION INVALID - YYY

Explanation: For the XXX operand, the operand was not enclosed in parentheses or specified an invalid register. YYY is the invalid notation.

System Action: The macro instruction was not expanded. Severity code=12.

Programmer Response: Probable user error. Correct the register notation or specify a valid register and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB079 FIRST OPERAND REGISTER NOTATION REQUIRED

Explanation: Register notation is required for the first operand.

System Action: The macro instruction was not expanded. Severity code=12.

Programmer Response: Probable user error. Specify a register notation for the first operand and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB080 ONE ECBLIST OPERAND ONLY REQUIRED

Explanation: The ECBLIST operand was omitted or more than one supplied.

System Action: The macro instruction was not expanded. Severity code=12.

Programmer Response: Probable user error, Supply one and only one ECBLIST operand and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB085 DEVD = xx CODED - EROPT = y IGNORED

Explanation: In a DCB macro instruction, one of the following occurred:

- Both DEVD=BS and EROPT=N were coded. However, EROPT=N is invalid for binary synchronous devices. Error recovery procedures are required.
- Both DEVD=WT and EROPT=R, W, or T were coded. However, EROPT=R, W, or T is invalid for World Trade telegraph terminals.

System Action: The macro instruction was expanded normally. The EROPT operand was ignored. Severity code= *.

Programmer Response: Probable user error. Remove the EROPT operand and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB100 X OR Y PARAMETER NOT WITHIN ALLOWABLE VALUE RANGE.

Explanation: In the ONLTST macro instruction, either the X or Y operand specified an incorrect value. The X operand must specify a value from 00 through 22, and the Y operand must specify a value from 01 through 99.

System Action: The macro instruction was not expanded. Severity code = 12.

Programmer Response: Probable user error. Correct the X or Y operand in the ONLTST macro instruction and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB103 TEXT OR LENGTH MISSING WHEN X = 0 OR X = 1.

Explanation: In the ONLTST macro instruction, although the X operand specified 0 or 1, either the TEXT or the LENGTH operand was missing. Whenever the X operand specifies 0 or 1, the TEXT and LENGTH operands must also be specified.

System Action: The macro instruction was not expanded. Severity code = 12.

Programmer Response: Probable user error. Include both the TEXT and LENGTH operands in the ONLTST macro instruction and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB104 TEXT OR LENGTH MISSING.

Explanation: In the ONLTST macro instruction, either the TEXT or the LENGTH operand was missing. If one of these two operands is specified, the other operand must also be specified.

System Action: The macro instruction was not expanded. Severity code = 1.

Programmer Response: Probable user error. Include both the TEXT and the LENGTH operands in the ONLTST macro instruction and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB105 X GREATER THAN 1. TEXT AND LENGTH PARAMETERS IGNORED.

Explanation: In the ONLTST macro instruction, although the X operand specified a value greater than 1, the TEXT and LENGTH operands were also specified. Whenever the X operand specifies a value greater than 1, the TEXT and LENGTH operands should not be specified.

System Action: The macro instruction is expanded normally, and the TEXT and LENGTH operands are ignored.

Programmer Response: Probable user error. Remove the TEXT and LENGTH operands from the ONLTST macro instruction. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB107 DIALCOUNT AND DIALCHARS NOT IN AGREEMENT.

Explanation: In the DFTRMLST macro instruction, the length of the telephone number specified in the dialcount operand is not the same as the number of dial digits specified in the dialchars operand.

System Action: The dial digits are generated as specified in the dialchars operand without regard to the length specified in the dialcount operand. Severity code = 4.

Programmer Response: Probable user error. Correct the dialcount or dialchars operand in error. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB108 POLLING CHARACTERS ARE IMPROPER.

Explanation: In the DFTRMLST macro instruction, the number of entries specified in the polling list was greater than 253 or one of the polling characters in an entry was hexadecimal FE, a value that must not be used as a polling character.

System Action: The macro instruction was not expanded. Severity code = 12.

Programmer Response: Probable user error. Correct the polling list. If the problem recurs, do the following before calling IBM for programming support:
- Have the associated program listing available.

IHB109 LENGTH OF POLLING CHARACTERS PER ENTRY IS IMPROPER.

Explanation: In the DFTRMLST macro instruction, the entries in the polling list are not all of the same length.

System Action: All entries are truncated or expanded to equal the length of the first entry. Severity code = 4.

Programmer Response: Probable user error. Correct the polling list so that all the entries are of the same length. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB110 DIAL CHARACTERS INVALID IN WTLIST.

Explanation: In a DFTRMLST macro instruction specifying a list type of WTLIST, dial digits were specified. However, a list type of WTLIST should be used only where manual dialing is intended, and no dial digits can be specified.

System Action: The macro instruction was not expanded. Severity code = 12.

Programmer Response: Probable user error. Remove the dial digits from the DFTRMLST macro and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB111 LENGTH OR ADDRESS OF TONE OMITTED.

Explanation: In a DFTRMLST macro instruction specifying an answering list of the WTLIST type, either the length or the address of the data tone characters was omitted. However, both operands must be included for a list of this type.

System Action: The macro instruction was not expanded. Severity code = 12.

Programmer Response: Probable user error. Make sure that both length and adress operands are specified for an answering list of the WTLIST type. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB113 IDCOUNT AND IDSENT DO NOT AGREE.

Explanation: In a DFTRMLST macro instruction, the value specified for the idcount operand does not equal the number of characters specified by the idsent operand.

System Action: The macro instruction was partially expanded; expansion stopped upon detection of the error. Severity code = 12.

Programmer Response: Probable user error. Correct the idcount value and reassemble. If the problem recurs, do the following before calling IBM for programming support:
- Have the associated program listing available.

IHB114 IDCOUNT IS TOO LARGE.

Explanation: In a DFTRMLST macro instruction, the value specified for idcount is greater than 16 (for a calling list of the AD or MD type), or is greater than 17 (for an answering list of the AN type).

System Action: The macro instruction was partially expanded; expansion stopped upon detection of the error. Severity code = 12.

Programmer Response: Probable user error. Correct the idcount value and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB115	INVALID TYPE ATTRIBUTES.

Explanation: In a DFTRMLST macro instruction, an invalid type attribute was specified for one of the operands.

System Action: The macro instruction was partially expanded; expansion stopped upon detection of the error. Severity code = 12.

Programmer Response: Probable user error. Correct the operand and reassemble. If the problem recurs, do the following before calling IBM for programming support.

- Have the associated program listing available

IHB116	AUTHORIZED SEQUENCE IS MISSING

Explanation: In a DFTRMLST macro instruction, either a control value or a user data area was specified without an authorized sequence having been specified.

System Action: The macro instruction was partially expanded; expansion stopped upon detection of the error. Severity code = 12.

Programmer Response: Probable user error. Either specify an authorized sequence or eliminate the control value or user data area. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

IHB117	PARENTHESIS IS MISSING.

Explanation: In a DFTRMLST macro instruction, the authorized sequence was not enclosed in parentheses.

System Action: The macro was partially expanded; expansion stopped upon detection of the error. Severity code = 12.

Programmer Response: Probable user error. Enclose the authorized sequence in parentheses and reassemble. If the problem recurs, do the following before calling IBM for programming support:

- Have the associated program listing available.

I/O ERROR MESSAGE

This message is printed at the console of the central computer following an error that BTAM error recovery procedures have failed to correct.

```
IEA000I aaa,I/O ERR,bb,cccc,ddee,ffgghhhh
```

IEA000I
: is the standard message code for the operator. The internal component name is IEA, the serial number is 000, and the action code is I (meaning information); immediate operator action is not required.

  The following information is typed in hexadecimal (except I/O ERR):

aaa
: is the address of the communication line on which the error occurred.

I/O ERR
: is the message text, indicating the occurrence of an I/O error.

bb
: is the command code of the failing command in the channel program. (See the DECCMCOD field in Appendix B for code values and meanings.)

cccc
: is the status bytes of the channel status word (CSW) as specified in the Input/Output Block (IOB).

dd
: is the first sense byte as specified in the IOB.

ee
: is the sense information resulting from issuing diagnostic Write or Read commands if the commands resulted in a unit check (IBM 2701 only).

ff
: is the TP operation code of the failing command in the channel program. (See the DECTPCOD field in Appendix B for code values and meanings.)

gg
: (not used)

hhhh
: is the terminal ID (polling or addressing characters). If only one polling character is used, it is left-justified in this field. (For IBM 2740 Model 2: When this message is issued for an addressing error, the first character (hh..) is the address of the terminal, and the second character (..hh) indicates the kind of error that occurred on the previous Write operation. The meanings of the codes are given in the IBM 2740 -- General Information section of the Start-Stop Read and Write Operations chapter. When the message is issued for a polling error, only one character, the polling character, appears at this point in the message.)

LINE ERROR RECORDING MESSAGES

These messages indicate the number of errors occurring for a given line.

Message IEC801I prints the contents of each of the four error threshold counters, indicating the number of data check, intervention required, or nontext time-out errors that have occurred since the last time the error threshold counters were reset. This message is printed whenever the threshold count has been reached for any of the three types of errors, or when the number of transmissions reaches the threshold count. (See the LERB (Line Error Recording Block) macro instruction for further information.)

Message IEC802I prints the contents of each of the four cumulative counters (accumulators), indicating the total number of data check, intervention required, and nontext timeout errors, and number of transmissions that have been accumulated since the cumulative counters were last reset. This message is printed whenever the user program issues a LERPRT macro instruction.

```
IEC801I aaa THRESHOLD TRANS=bbb DC=ccc IR=ddd TO=eee
```

IEC801I
: is the standard message code for the operator. The internal component name is IEC, the serial number is 801, and the action code is I, meaning information; immediate operator action is not required.

aaa
: is the address of the communication line on which the error occurred (printed in hexadecimal).

THRESHOLD
: is the message text.

TRANS=bbb
: is the number of transmissions that have occurred on this line (in decimal).

DC=ccc
: is the number of data check errors that have occurred on the line during the indicated number of transmissions (in decimal).

IR=ddd
: is the number of intervention required errors that have occurred on the line during the indicated number of transmissions (in decimal).

TO=eee
: is the number of nontext timeout errors that have occurred on the line during the indicated number of transmissions (in decimal).

```
IEC802I aaa LINE TOTALS  TRANS=bbbbbbbb DC=ccccc IR=ddddd TO=eeeee
```

IEC802I
: is the standard message code for the operator. The internal component name is IEC, the serial number is 802, and the action code is I, meaning information; immediate operator action is not required.

aaa
: is the address of the communications line on which the errors occurred.

LINE TOTALS
: is the message text, indicating the total number of errors on the specified line.

TRANS=bbbbbbbb
: is the total number of transmissions that have occurred on the line since the accumulators were reset (in decimal).

DC=ccccc
: is the total number of data check errors that have occurred on the line during the indicated number of transmissions (in decimal).

IR=ddddd
: is the total number of intervention required errors that have occurred on the line during the indicated number of transmissions (in decimal).

TO=eeeee
: is the total number of nontext time-out errors that have occurred on the line during the indicated number of transmissions (in decimal).

TRANSMISSION CONTROL UNIT INOPERATIVE MESSAGE

This message is issued whenever an IBM 2701, 2702, or 2703 becomes inoperative. Usually, this message is printed during opening of a line group associated with the inoperative TCU. It appears when an I/O operation for some line connected to that TCU is attempted. When the message appears, the central computer operator should determine the reason for the condition and reactivate the TCU.

```
IEC804A aaa CONTROL UNIT NOT OPERATIONAL.  REPLY
            CONT OR POST
```

IEC804A
: is the standard message code for the operator. The internal component name is IEC, the serial number is 804, and the action code is A, meaning operator action is required.

aaa
: is the address of the communications line.

CONTROL UNIT NOT OPERATIONAL. REPLY CONT OR POST
: is the message text, indicating the response for the operator. Reply either CONT or POST. If the operator replies CONT, the I/O operation for which this message was printed will be retried. If the retry is unsuccessful, the message will be reissued. If successful, the operation will continue. If the reply is POST, the operation will be posted complete-with-error and the "not operational SIO" bit (bit 0) will be turned on in the DECERRST field of the DECB for the line.

Note: If a reply is not entered before the requesting job is cancelled, the system may enter wait state.

IEC809I aaa CONTROL UNIT NOT OPERATIONAL

IEC809I
is the standard message code for the operator. The internal component name is IEC, the serial number is 809, and the action code is I, meaning no operator action is required.

aaa
is the address of the communications line.

CONTROL UNIT NOT OPERATIONAL
is the message text.

REMOTE BSC STATION ERROR MESSAGES

BTAM allows remote BSC stations to send error information to the central computer.* This information is routed to the user program, to the central computer console, or to an error file on a system residence device. The format of the error message depends on which of these destinations is desired:

1. Error information to be sent to the user program must appear in a message having this format:

SOH	%	S[1]	DLE[2]	STX	text[4]	ETX
1	1	1	1	1		1

2. Error information to be sent to the central computer console must appear in a message having this format:

SOH	%	C[1]	DLE[2]	STX	r[3]	text[4]	ETX
1	1	1	1	1	1		1

3. Error information to be sent to the system error file must appear in a message having this format:

SOH	%	E[1]	DLE[2]	STX	X'08'	reserved	ID	text[4]	ETX
1	1	1	1	1	1	2	2		1

4. **Error information from a remote 3270 display system to be sent to the problem program and to be recorded as T-type records in SYS1.LOGREC must appear in a message having this format:**

SOH	%	R[1]	STX	text[4]	ETX
1	1	1	1	4	1

Notes:

[1]This character must be uppercase (EBCDIC or USASCII).

[2]DLE need be present only for transparent text; however, in the third format (system error file), the text must begin in the eleventh byte.

[3]r is the routing code that specifies the console to which this message is to be routed.

[4]The text of each message depends on the format:
Format 1 - The text is user provided; its length depends on the size of the user's buffer.
Format 2 - The text must consist of printable characters. The length of the text must be either 17 or 60 characters; extra characters will be automatically truncated.
Format 3 - The text is bit significant. The length may be from 35 to 210 characters, but must be a multiple of 35 (i.e., it can be 35, 70, 105, 140, 175, or 210 characters). If necessary, the text should be padded with 'FF' bytes.

Format 4 - The text includes the control unit and device addresses of the remote 3270 device from which the message was received and two sense/status bytes.

After sending the error message to the user program, console, or error file, BTAM restarts the user-program Read operation with which the error message was received. The remote station then may send another error message, a regular message, or EOT.

Upon receiving an error message with a Read Initial Operation on a multipoint line, BTAM preserves the Auto Poll index byte in the first byte of the input area. Therefore, following each Read Initial operation on a multipoint line, the user program should check the second byte of the input area for an EOT character.

***Currently, the BSC stations that can send these messages are the 2715 (second and third formats only), the 2770 (first format only), and the remote 3270 (fourth format only).**

ERROR STATUS MESSAGES (IBM 2770)

One of the following five error messages may be sent by the 2770 terminal operator, as specified by the error recovery procedure for the 2770.

- Checkpoint Restart:
  Last Restart Point: SOH % S STX 0 X_1 X_2 C SP SP
  Specific Restart Point: SOH % S STX 2 X_1 X_2 C text...

(The first format causes BTAM to begin retransmission at the point from which the previous transmission began, or at any other point decided by the user program when it detects the 'C' preceding the two space characters. The second format allows the terminal operator to indicate to the user program where he wishes for retransmission to begin. The text can be up to 50 characters long and can contain any information the user program needs to identify the point at which transmission is to begin. This might be, for example, a page number or form number.

- Customer Engineer Attention Required: SOH % S STX 0 X_1 X_2 D Z_1 Z_2
- Job Restart: SOH % S STX 0 X_1 X_2 M SP SP
- Format Error: SOH % S STX 0 X_1 X_2 F SP SP

In these formats:

X_1 is the station address

X_2 is the component address

text is any information the terminal operator wishes to send to identify to the user program the point from which retransmission is to begin.

Z_1 is the station address. This is the same as X_1 if the component requires attention by a Customer Engineer, but the station is operational; it is the address of an alternate station if the sending station requires CE attention; and it is a SP character if no alternate station is available or desired.

Z_2 is the component address of an alternate componentatthe sending station or at an alternate station.

When BTAM recognizes an error status message, it posts the operation complete with a completion code of X'7F' and turns on bit 0 of DECFLAGS.

TERMINAL ERROR STATUS MESSAGE (IBM 2715)

This message, in one of four formats, provides the results of a scan of the error file of an IBM 2715 Transmission Control. The scan occurs when the error threshold for one of the area stations connected to the 2715 is exceeded (threshold value is eight) or when manually requested at the 2715, the 2740 attached to the 2715, or the central computer. BTAM prints the message on the master console, the teleprocessing console, or the system maintenance console, depending on the routing code included in the error scan message sent by the 2715. (The routing code does not appear in the message printed on the console.)

In the four formats below:

cuu
: is the address of the communications line (channel and unit) (EBCDIC).

xx
: is the address of the area station for which the error scan is reported (hexadecimal).

tttt
: is the time (0001-2400) the error scan occurred (decimal).

ww
: is the address of a particular adapter within the 2715 (hexadecimal).

Other fields in the message are indicated under individual formats below.

```
IEC815I cuu  xx tttt yy ERS z
```

Explanation: This message reports the results of an error scan by the 2715 when five or more of the eight errors involved a particular one of the devices attached to the area station.

yy
: is the address of the device for which the errors occurred (hexadecimal).

z
: is the number of errors (from 5 to 8) that occurred for the device (decimal).

Operator Response: None.

```
IEC815I cuu  xx tttt THRESHLD
```

Explanation: This message indicates that the threshold value of eight has been reached for the area station whose address is xx, but no one device attached to the station accounted for as many as five of the errors.

Operator Response: None

```
IEC815I cuu  xx tttt yy eeee zzzz yy eeee zzzz yy eeee zzzz yy eeee zzzz
```

Explanation: This message is issued whenever an error scan for a particular area station is manually requested at the 2715, 2740 attached to 2715, or the central computer. The message appears twice in succession. Each indicates the nature of four errors; the two messages together provide this information for the eight most recent occurences for area station xx.

yy
: is the address of a device (hexadecimal).

eeee
: is the error data for device yy (hexadecimal).

zzzz
is the time (0001-2400) the error data was recorded on the 2715 disk (decimal).

Operator Response: None.

```
IEC815I cuu ww tttt eeeeeee zzzz eeeeeee zzzz eeeeeee zzzz eeeeeee zzzz
```

Explanation: This message is issued whenever an error scan for a particular 2715 adapter is manually requested at the 2715, 2740 attached to 2715, or the central computer. The message appears twice in succession. Each indicates the nature of four errors; the two messages together provide this information for the eight most recent error occurrences for adapter ww.

eeeeeee
is the error data for adapter ww (hexadecimal).

zzzz
is the time (0001-2400) the error data was recorded on the 2715 disk (decimal).

Operator Response: None.

ERROR STATUS MESSAGE (REMOTE IBM 3270)

An error status message from a remote 3270 device has the format:

SOH	%	R	STX	CU address	device address	sense/status byte 1	sense/status byte 2	ETX

cu address — is the address of the control unit of the device from which the message was received (see Figure 3270-1 in the section "IBM 3270 Display System - Programming Considerations").

device address — is the address of the device from which the message was received (see Figure 3270-3 in the section "IBM 3270 Display System - Programming Considerations").

sense/status byte 1 has the format:

Bit	Definition
0	Setting depends on bits 2-7 (see Figure 3270-4 in the section "IBM 3270 Display System - Programming Considerations")
1	Setting always 1
2	Reserved
3	Reserved
4	Device Busy (DB)
5	Unit Specify (US)
6	Device End (DE)
7	Transmission Check (TC)

sense/status byte 2 has the format:

Bit	Definition
0	Setting depends on bits 2-7 (see Figure 3270-4)
1	Setting always 1
2	Command Reject (CR)
3	Intervention Required (IR)
4	Equipment Check (EC)
5	Data Check (DC)
6	Control Check (CC)
7	Operations Check (OC)

For more information about the sense/status bytes, see <u>IBM 3270 Information Display System, Component Description</u>, GA27-2749.

When BTAM recognizes an error status message from a remote 3270 device, the operation is posted with a completion code of X'7F', and bits 1 and 6 are turned on in the DECFLAGS field of the DECB. BTAM sends the error status message to the problem program and records the information as T-type records in SYS1.LOGREC.

Error Condition			Suggested Action
Sense/Status Bytes	Bit(s) Set	Unit(s)	
X'4050'	IR	3271, 3275	6
X'4060'	CR	3271, 3275	8
X'40C1'	OC	3271, 3275	8
X'40C2'	CC	3271	4
X'40C3'	CC, OC	3271	2
X'40C4'	DC	3271, 3275	4
X'40C6'	DC, OC	3271	2
X'40D1'	IR, OC	3271	7
X'4C40'	DB, US	3271, 3275	14
X'4E40'	DB, US, DE	3271, 3275	4
X'C140'	TC	3275	16
X'C240'	DE	3271, 3275	NA
X'C250'	IR, DE	3271, 3275	6
X'C2C4'	DC, DE	3271, 3275	4
X'C2C8'	EC, DE	3275	11
X'C2D8'	IR, EC, DE	3275	11
X'C4C1'	OC, US	3271	17
X'C4C4'	DC, US	3271, 3275	4
X'C4C5'	DC, OC, US	3271	5
X'C6C4'	DC, US, DE	3271, 3275	12
X'C6C8'	EC, US, DE	3271	11
X'C6D8'	TR, EC, US, DE	3271	11
X'C840'	DB	3271, 3275	13
X'C8C1'	DB, OC	3271	15

Table 47C. Suggested Actions According to Remote 3270 Error Status Message

Table 47C indicates suggested actions according to the contents of the sense/status bytes in error status messages. The suggested actions are:

1. Execute a new address selection sequence, and retransmit the message starting with the command sequence that was being executed when the error occurred. If the operation is not successful after two retries, consider the error nonrecoverable, and take action 9.
2. Do the same as in action 1, except take action 10 after two retries.
3. Do the same as in action 1, except retransmit the entire failing chain of commands.
4. If possible, reconstruct the entire screen buffer image, and retry the failing chain of commands (within the BSC sequence of operations). If the screen buffer cannot or need not be reconstructed, retry the operation anyway. If the operation is not successful after three retries, consider the error nonrecoverable, and take action 9.
5. Do the same as in action 4, except reconstruct the buffer of the "from" device specified in the copy command. If the operation is not successful after three retries, consider the error nonrecoverable, and take action 10.
6. Wait for the display operator or system operator to ready the printer. Retry the printout by issuing a write command with the WCC and no data stream. Or take action 4.
7. Wait for the display operator or system operator to ready the "from" device specified in the copy command. Take action 2.
8. Examine the data stream to determine the cause of the nonrecoverable programming error.
9. Request maintenance on the malfunctioning device. After repair, try to reconstruct the screen buffer image (using an erase/write command to correct a missing or multiple cursor condition in the buffer). Retry the failing chain of commands as in the previous action.
10. Request maintenance on the malfunctioning device (the "from" device specified in the copy command). After repair, try to reconstruct the screen buffer image (using an erase/write command to correct a missing or multiple cursor condition in the buffer). Retry the failing chain of commands as in the previous action.
11. If a new printout is required, take action 6.
12. If a new printout is required, take action 4.
13. Periodically issue a specific poll to read the Device End indication that is sent by the device to the TCU when the device goes not busy.
14. Periodically issue a specific poll to read the Device End indication that is sent by the device to the TCU when the device goes not busy. Take action 4.
15. Do the same as in action 14, except take action 1 when the "from" device specified in the copy command goes not busy.
16. If the failing command is (1) a write command with a data stream of more than one byte or (2) one of a chain commands that contains a previous write command without an SBA order immediately following the WCC, take action 4. Otherwise, take action 3.
17. An unauthorized attempt was made to copy data from a device. The device address in the error status message is the address of the "to" device specified in the copy command.

090 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.

An Open routine found that a device other than a communications device was allocated to the data control block (DCB) being opened; that is, the device class code in the unit control block (UCB) for the device allocated to the data control block was not equal to hexadecimal 40.

Programmer Response: Either the UNIT parameter of the DD statement for the communications device is incorrect or unit control block generated during system generation is invalid. Check for improper specification of the UNIT parameter of the DD statement or the UNIT operand of the IODEVICE macro instruction. After correcting the error, execute the job step again. If the problem recurs, do the following before calling IBM for programming support:

- Make sure that MSGLEVEL=(1,1) was specified in the JOB statement and that a SYSABEND DD statement was included for the failing job step.
- Have the associated job stream and program listing available.

091 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.

An Open routine found an invalid or unsupported type of transmission control unit specified in the unit control block (UCB) for the device allocated to the data control block (DCB) being opened.

Programmer Response: Check for improper specification of the IOCONTRL macro instruction used in generating the system. After correcting the error, execute the job step again. If the problem recurs, do the following before calling IBM for programming support:

- Make sure that MSGLEVEL=(1,1) was specified in the JOB statement, and that a SYSABEND DD statement was included for the failing job step.

- Have the associated job stream and program listing available.

092 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.

An Open routine found an invalid or unsupported type of terminal control or terminal adapter specified in the unit control block (UCB) for the device allocated to the data control block (DCB) being opened.

Programmer Response: Check for improper specification of the ADAPTER operand in the IODEVICE macro instruction used in generating the system. Correct the error and execute the job step again. If the problem recurs, do the following befor calling IBM for programming support:

- Make sure that MSGLEVEL=(1,1) was specified in the JOB statement, and that a SYSABEND DD statement was included for the failing job step.

- Have the associated job stream and program listing available.

093 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.

An Open routine found an invalid or unsupported type of terminal specified in the unit control block (UCB) for the device allocated to the data control block (DCB) being opened.

Programmer Response: Check for improper specification of the UNIT operand in the IODEVICE macro instruction used in generating the system. Correct the error and execute the job step again. If the problem recurs, do the following before calling IBM for programming support:

- Make sure that MSGLEVEL=(1,1) was specified in the JOB statement and that a SYSABEND DD statement was included for the failing job step.

- Have the associated job stream and program listing available.

094 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.

An Open routine found an invalid or unsupported optional feature or mode of operation specified in the unit control block (UCB) for the device allocated to the data control block (DCB) being opened.

Programmer Response: Check for improper specification of the FEATURE operand in the IODEVICE macro used in generating the system. Correct the error and execute the job step again. If the problem recurs, do the following before calling IBM for programming support:

- Make sure that MSGLEVEL=(1,1) was specified in the JOB statement and that a SYSABEND DD statement was included for the failing job step.

- Have the associated job stream and program listing available.

095 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.

An Open routine found that the lines allocated to the line group did not have identical types of terminals or lines, or that the terminals did not have the identical features.

Programmer Response: Determine which line group contains different types of terminals or lines and redefine its lines through DD statements or a new system generation. If the problem recurs, do the following before calling IBM for programming support:

- Make sure that MSGLEVEL=(1,1) was specified in the JOB statement and that a SYSABEND DD statement was included for the failing job step.
- Have the associated job stream and program listing available.

096 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.

An Open routine found that dynamic buffer allocation had been specified in the DCBBFTEK field of the data control block (DCB). However, the Open routine could not dynamically allocate buffers because the data control block specified neither the address of a buffer pool control block (in the DCBBUFCB field) nor the number and length of the buffers (in the DCBBUFNO and DCBBUFL fields).

Programmer Response: Correct the error by (1) providing a buffer pool and specifying the address of its control block in the DCBBUFCB field, (2) specifying the number and length of the buffers in the DCBBUFNO and DCBBUFL fields, or (3) handling buffering in the user program and deleting the BFTEK=D operand in the DCB macro instruction or the DCB parameter of the DD statement. Then execute the job step again. If the problem recurs, do the following before calling IBM for programming support:

- Make sure that MSGLEVEL=(1,1) was specified in the JOB statement and that a SYSABEND DD statement was included for the failing job step.
- Have the associated job stream and program listing available.

097 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.

An Open routine required an additional entry in the device I/O directory; however, the directory was already full. Since the last system start, the maximum number of device types have been allocated. Normally, the maximum number is 16.

Programmer Response: Do the following before calling IBM for programming support:

- Make sure that MSGLEVEL=(1,1) was specified in the JOB statement and that a SYSABEND DD statement was included for the failing job step.
- Have the associated job stream and program listing available.

098 Explanation: The error occurred during execution of a BTAM OPEN macro instruction.

Although Dual Communication Interface B or Dual Code Feature B was specified in the data control block (DCB), the transmission control unit is not an IBM 2701 or the unit control block (UCB) established at system generation time did not specify that the 2701 is equipped with the Dual Communication Interface or Dual Code feature.

Programmer Response: Probable user error. Correct the DCB macro that defined the data control block that erroneously specified the Dual Communication Interface B or Dual Code B, reassemble, and re-execute the job step. If the problem recurs, do the following before calling IBM for programming support:

- Make sure that MSGLEVEL=(1,1) was specified in the JOB statement and that a SYSABEND DD statement was included for the failing job step.
- Have the associated job stream and program listing available.

This appendix explains the operands that must be included in the system generation macro instructions when generating an operating system that includes BTAM. Only those macro instructions and operands directly related to BTAM are given; for other macro instructions required for generating a system, and an explanation of the generation process, see the OS SYSGEN publication. In particular, refer to explanations of the DATAMGT, IOCONTRL, IODEVICE, and TELCMLIB macro instructions.

TELCMLIB Macro Instruction

During system generation, you must specify the TELCMLIB macro instruction. It causes the telecommunications subroutine library, SYS1.TELCMLIB, to be included in your operating system.

DATAMGT Macro Instruction

DATAMGT causes one or more optional access methods to be included in your operating system. To include BTAM, code:

Name	Operation	Operand
	DATAMGT	ACSMETH=BTAM

IOCONTRL Macro Instruction

IOCONTRL identifies to the operating system the type of transmission control unit (TCU) or control unit to be attached to a S/360 channel control unit address. Specify one IOCONTRL macro for each TCU or control unit to be operated under BTAM. Only the operands shown are applicable for a BTAM system.

If you wish to specify an IBM 2702 having the 31-line expansion feature, you must code a separate IOCONTRL macro for each of the two sets of lines.

Name	Operation	Operand
[name]	IOCONTRL	UNIT=type, ADDRESS=address MODEL=model,

UNIT
: Specifies the type of transmission control unit: 2701, 2702, or 2703. For the local 3270 display system, specifies the type of control unit: 3272.

ADDRESS
: Specifies the control unit address to which the TCU is connected. For the local 3270 display system, specifies the channel control unit address to which the 3272 control unit is connected. The address value consists of two hexadecimal digits having a valid range of 00 to 6F. The first digit identifies the channel, and the second identifies the control unit address.

MODEL
: Specifies whether the 3272 control unit given by the UNIT operand is a model 1 (480-character buffer) or model 2 (1920-character buffer). Code MODEL=1 or MODEL=2.

Note: The MODEL operand applies only to the local 3270 display system.

IODEVICE Macro Instruction

IODEVICE describes to the operating system the characteristics of an input/output device and its operating system requirements. For BTAM, IODEVICE identifies the type of device, i.e., remote station, that is connected to a communications line, or the type of line configuration. You therefore code one IODEVICE macro for each line, regardless of how many remote stations are connected to the line. For BTAM support for the local 3270 display system, IODEVICE identifies the type of local 3270 device connected to a 3272 control unit. You code one IODEVICE macro for each local 3270 device.

Only the operands shown are applicable for a BTAM system.

Name	Operation	Operand
symbol	IODEVICE	UNIT=type ADDRESS=address MODEL=model, ADAPTER=type [,FEATURE=(feature1, feature2, . . .)] [,SETADDR=type] [,OBRCNT=n]

UNIT
: specifies the type of remote terminal (start-stop) or type of line configuration (BSC) associated with the line address given by the ADDRESS operand. For the local 3270 display system, specifies the type of local device with the device address given by the ADDRESS operand. VAlid UNIT parameters are:

- For start-stop:

  1030
  1050
  1060
  2260
  2740
  2741P (2741 using PTTC code)
  2741C (2741 using Correspondence code)
  83B3
  115A
  TWX
  WTTA (World Trade Telegraph Terminals)

- For BSC

  BSC1 (for nonswitched point-to-point line)
  BSC2 (for switched point-to-point line)
  BSC3 (for nonswitched multipoint line)

- For local 3270 display system:

  3277
  3284
  3286

Alternatively, for BSC, specific station types may be coded: S360 (S/360 except Model 20), 2020 (S/360 Model 20), 1130, and 2780. If more than one of these types of station are connected to the same multipoint line, or can call the central computer over the same switched line termination (telephone number), the following rules apply:

- If the combination consists only of the S/360 and S/360 Model 20, code UNIT=2020.
- If the combination includes 2780, together with S/360 or S/360 Model 20, or both, but excluding 1130, code UNIT=2780.
- If the combination includes 1130, code UNIT=1130, regardless of which other types are in the combination.

These alternative UNIT values provide compatibility with earlier versions of BTAM, that is, the UNIT operand need not be recoded as BSC2 or BSC3. Future releases of BTAM will, however, require that BSC1, BSC2, or BSC3 be coded.

ADDRESS
: Specifies the three-digit address of the line over which the type of station given by UNIT is to communicate. For the local 3270 display system, specifies the three-digit address of the local 3270 device given by the UNIT operand. Valid parameters are within the range 000-6FF, inclusive (hexadecimal).

MODEL
: Specifies whether the local 3270 device given by the UNIT operand is a model 1 (480-character buffer) or model 2 (1920-character buffer). Code MODEL=1 or MODEL=2.

Note: The MODEL operand applies only to the local 3270 display system. Only model 1 devices (3277, 3284, 3286) may be connected to a model 1 3272 control unit. Model 1 or model 2 devices or both may be connected to a model 2 3272 control unit.

ADAPTER
: Specifies the type of TCU terminal control and terminal adapter associated with the line address given by the ADDRESS operand. Code one of the following values:

IBM1 For IBM 1050,1060, 2740 or 2741 communicating with:

- IBM 2701 through an IBM Terminal Adapter, Type I, and either: (1) an appropriate data set, or (2) an IBM Line Adapter.
- IBM 2702 or 2703 through an IBM Terminal Control, Type I, and either: (1) a Data Set Line Adapter and an appropriate data set, or (2) an IBM Line Adapter.

IBM2 For IBM 1030 communicating with:

- IBM 2701 through an IBM Terminal Adapter, Type II, and either: (1) an appropriate data set, or (2) an IBM Line Adapter.
- IBM 2702 or 2703 through an IBM Terminal Control, Type II, and either: (1) a Data Set Line Adapter and an appropriate data set, or (2) an IBM Line Adapter.

IBM3 For IBM 2260-2848 communicating with:

- IBM 2701 through an IBM Terminal Adapter, Type J˜T and an appropriate data set.

IBMT For IBM 1050 communicating with:

- IBM 2701 through an IBM Telegraph Adapter.
- IBM 2703 through an IBM Terminal Control, Type I, and a Telegraph Line Adapter.

TELE1 For AT&T 83B3 or Western Union 115A communicating with:

- IBM 2701 through a Telegraph Adapter, Type I.
- IBM 2702 or 2703 through a Telegraph Terminal Control, Type I, and a Telegraph Line Adapter.

TELE2 For WU TWX (Model 33 or 35) communicating with:

- IBM 2701 through a Telegraph Adapter, Type II, and an appropriate data set.
- IBM 2702 or 2703 through a Telegraph Terminal Control, Type II, and a Data Set Line Adapter and an appropriate data set.

TELEW For World Trade Telegraph terminal communicating with:

- IBM 2701 through a World Trade Telegraph Adapter.
- IBM 2702 or 2703 through a World Trade Telegraph Adapter and a Telegraph Line Adapter.

BSCA For IBM System/360, System/360 Model 20, System/3, 1130, 1800, 2715, 2770, 2780, or 2972 communicating with:

- IBM 2701 through a Synchronous Data Adapter, Type II, and an appropriate data set.
- IBM 2703 through a Synchronous Terminal Control and an appropriate data set.

FEATURE
Specifies certain optional features with which the transmission control unit (TCU) or remote station is equipped. For the local 3270 display system, specifies certain features with which the local 3270 display station is equipped. Code each of the applicable parameters:

AUTOCALL if the TCU (2701, 2702, or 2703) to which the remote station is connected is equipped with the Auto Call feature and the line is connected to the TCU terminal adapter by means of an Automatic Calling Unit and an appropriate data set. When these conditions are met, and you specify AUTOCALL, the channel programs generated for the line whose address is specified by the ADDRESS operand can automatically dial the remote stations, using the telephone number you specify in the terminal list associated with the line.

AUTOANSR if the data set (modem) connecting the access line specified by the address operand to the TCU is a switched line over which calls are to be answered. AUTOANSR must be coded regardless of whether the line is equipped with an automatic answering unit. When you specify AUTOANSR, the channel programs generated for the specified line will automatically initiate message transmission when a remote station calls the computer.

AUTOPOLL if the automatic polling facility of the TCU is to be used. This facility is a standard feature of the IBM 2703, and an optional feature (called Auto Poll) of the IBM 2702. For the IBM 2701 this parameter is valid only for lines connected through the Synchronous Data Adapter Type II. If you specify AUTOPOLL, the Read Initial channel programs generated for the specified line will be so arranged that a negative response from a remote station causes the TCU to automatically poll the next station in the terminal list without signalling an I/O interrupt. If you omit AUTOPOLL, Read Initial operations will employ programmed polling with each negative response from a remote station causing an I/O interrupt. Only those Read Initial operations that send polling characters are affected.

This parameter is valid only for nonswitched multipoint lines to which are connected the IBM 1030, 1060, 1050, 2740, or any BSC stations, (as specified by the UNIT operand), as only these types of stations can be polled using the Auto Poll facility. The AUTOPOLL operand <u>must</u> be coded for BSC stations on multipoint lines, and <u>may</u> be coded for the foregoing start-stop terminals.

If UNIT=2740 is specified, you must also code in the FEATURE operand, either SCONTROL or SCONTROL and CHECKING (in addition to the AUTOPOLL parameter).

DUALCOMM if the TCU (IBM 2701 only) to which the line specified by ADDRESS is connected is equipped with the Dual Communication Interface special feature. This feature allows program selection (in the DCB macro) of either of two data sets (modems) over which transmission is to occur (BSC lines only).

DUALCODE if the TCU (IBM 2701 only) is equipped with the Dual Code special feature. This feature allows program selection (in the DCB macro) of the transmission code to be used on the communication line (BSC lines only).

<u>For IBM 2740 Terminals only:</u>

CHECKING if UNIT=2740 is specified and the terminal is equipped with the Record Checking special feature.

SCONTROL if UNIT=2740 is specified and the terminal is equipped with the Station Control special feature. This parameter and the AUTOCALL, AUTOANSR, OIU, and XCONTROL parameters are mutually exclusive.

XCONTROL if UNIT=2740 is specified and the terminal is equipped with the Transmit Control special feature and the Dial Up special feature. You also must indicate the Dial Up special feature in the FEATURE operand by the AUTOCALL or AUTOANSR parameter, or both, as appropriate. The XCONTROL parameter and the OIU parameter are mutually exclusive.

OIU if the UNIT=2740 is specified and the terminal is equipped with an IBM 2760 Optical Image Unit. This parameter and the SCONTROL and XCONTROL parameters are mutually exclusive.

<u>For local 3270 devices only</u>:

- One of the following character generator options:

  DOCHAR if the device has a domestic monocase character generator. If the FEATURE operand is not coded, this value is assumed.

  ASCACHAR if the display station has an ASCII A monocase character generator.

  ASCBCHAR if the display station has an ASCII B monocase character generator.

  FRCHAR if the device has a French monocase character generator.

  GRCHAR if the device has a German monocase character generator.

  KACHAR if the device has a Katakana monocase character generator.

  UKCHAR if the device has a United Kingdom monocase character generator.

- One of the following keyboard options (if a keyboard is present):

  EBKY3277 if the display station has an EBCDIC typewriter keyboard.

  ASKY3277 if the display station has an ASCII typewriter keyboard.

  DEKY3277 if the display station has a data entry keyboard.

  OCKY3277 if the display station has an operator console keyboard.

- One of the following keyboard options (if a keyboard is present):

  KB66KEY if the display station has a 66-key keyboard (that is, has no program function keys).

  KB78KEY if the display station has a 78-key keyboard (that is, has program function keys).

  KB70KEY if the display station has a Katakana character generator and a 70-key data entry keyboard.

  KB81KEY if the display station has a Katakana character generator and an 81-key EBCDIC typewriter keyboard.

- As many of the following options as required:

  SELPEN if the display station has a selector pen.

  NUMLOCK if the display station has the numeric lock feature.

  AUDALRM if the display station has a keyboard and an audible alarm.

  MAGCDRD if the display station has a magnetic card reader adapter.

SETADDR

Specifies which of the four Set Address (SAD) commands is to be issued to the transmission control unit (IBM 2702 only) for operations on the line specified by the ADDRESS operand. The SAD command selects the appropriate line speed for the type of terminal connected to the line. The association between the specific command (Sadzer, Sadone, Sadtwo, or Sadthree) and the corresponding line speed is established by internal connections within the 2702; this is done by the customer engineer when the 2702 is installed. You must code this operand if the TCU to which the line is connected is a 2702; if it is a 2703, the SAD commands will be ignored. Do not code this operand if the TCU is a 2701, as a command reject will be signalled when the line group is opened.

Code:	If the SAD command for the line is:
0	Sadzer
1	Sadone
2	Sadtwo
3	Sadthree

For IBM 2715 Transmission Control Unit only:

OBRCNT

specifies the number of area stations connected to the 2715s on the line represented by the IODEVICE macro. (This value is used to compute the space required on SYS1.LOGREC for error data received from the 2715.)

APPENDIX E: CODE CHARTS FOR BINARY SYNCHORONOUS COMMUNICATION AND THE LOCAL 3270 DISPLAY SYSTEM

Six Bit Transcode

Code Positions 2 3 4 5	0 0	0 1	1 0	1 1
0 0 0 0	SOH 12-9-1	& 12	- 11	0 0
0 0 0 1	A 12-1	J 11-1	/ 0-1	1 1
0 0 1 0	B 12-2	K 11-2	S 0-2	2 2
0 0 1 1	C 12-3	L 11-3	T 0-3	3 3
0 1 0 0	D 12-4	M 11-4	U 0-4	4 4
0 1 0 1	E 12-5	N 11-5	V 0-5	5 5
0 1 1 0	F 12-6	O 11-6	W 0-6	6 6
0 1 1 1	G 12-7	P 11-7	X 0-7	7 7
1 0 0 0	H 12-8	Q 11-8	Y 0-8	8 8
1 0 0 1	I 12-9	R 11-9	Z 0-9	9 9
1 0 1 0	STX 12-9-2	SPACE No Punch	ESC 0-9-7	SYN 9-2
1 0 1 1	. 12-8-3	$ 11-8-3	, 0-8-3	# 8-3
1 1 0 0	⌑ 12-8-4	* 11-8-4	% 0-8-4	@ 8-4
1 1 0 1	BEL 9-7	US 11-9-8-7	ENQ 0-9-8-5	NAK 9-8-5
1 1 1 0	SUB 9-8-7	EOT 0-9-8-7	ETX 12-9-3	EM 11-9-8-1
1 1 1 1	ETB 0-9-6	DLE 12-11-9-8-1	HT 12-9-5	DEL 12-9-7

Standard Representation of USASCII

Rows	Columns	0	1	2	3	4	5	6	7
	$b_7b_6b_5$	000	001	010	011	100	101	110	111
	$b_4b_3b_2b_1$								
0	0 0 0 0	NUL	DLE	SP	0	@	P	`	p
1	0 0 0 1	SOH	DC1	!	1	A	Q	a	q
2	0 0 1 0	STX	DC2	"	2	B	R	b	r
3	0 0 1 1	ETX	DC3	#	3	C	S	c	s
4	0 1 0 0	EOT	DC4	$	4	D	T	d	t
5	0 1 0 1	ENQ	NAK	%	5	E	U	e	u
6	0 1 1 0	ACK	SYN	&	6	F	V	f	v
7	0 1 1 1	BEL	ETB	'	7	G	W	g	w
8	1 0 0 0	BS	CAN	(	8	H	X	h	x
9	1 0 0 1	HT	EM	)	9	I	Y	i	y
10	1 0 1 0	LF	SUB	*	:	J	Z	j	z
11	1 0 1 1	VT	ESC	+	;	K	[	k	{
12	1 1 0 0	FF	FS	,	<	L	\	l	¦
13	1 1 0 1	CR	GS	-	=	M	]	m	}
14	1 1 1 0	SO	RS	.	>	N	^	n	~
15	1 1 1 1	SI	US	/	?	O	_	o	DEL

Data Link Control Functions

Function	Characters Used In: EBCDIC	USASCII	TRANSCODE
ACK-0	DLE, X'70'	DLE, 0	DLE, - (hyphen)
ACK-1	DLE, X'61'	DLE, 1	DLE, T
WACK	DLE, X'6B'	DLE, ;	DLE, Z
RVI	DLE, X'7C'	DLE, <	DLE, 2

EBCDIC

Bit Positions 4, 5, 6, and 7 ↓ / Bit Positions 0 and 1 →	00				01				Digit Punches
Bit Positions 2 and 3 →	00	01	10	11	00	01	10	11	
0000	① NUL	② DLE	③ DS	④	⑤ SP	⑥ &	⑦ -	⑧	
0001	SOH	DC1	SOS				⑬ /		1
0010	STX	DC2	FS	SYN					2
0011	ETX	DC3							3
0100	PF	RES	BYP	PN					4
0101	HT	NL	LF	RS					5
0110	LC	BS	EOB	UC					6
0111	DEL	IL	PRE	EOT					7
1000		CAN							8
Zone Punches	9 12	9 11	9 0	9	9 12 0	9 12 11	9 11 0	9 12 11 0	

Bit Positions 4, 5, 6 and 7 ↓ / Bit Positions 0 and 1 →	10				11				Digit Punches
Bit Positions 2 and 3 →	00	01	10	11	00	01	10	11	
0000					⑨	⑩	⑪	⑫ 0	8-1
0001	a	j			A	J	⑭	1	1
0010	b	k	s		B	K	S	2	2
0011	c	l	t		C	L	T	3	3
0100	d	m	u		D	M	U	4	4
0101	e	n	v		E	N	V	5	5
0110	f	o	w		F	O	W	6	6
0111	g	p	x		G	P	X	7	7
1000	h	q	y		H	Q	Y	8	8
1001	i	r	z		I	R	Z	9	9
Zone Punches	12 0	12 11	11 0	12 11 0	12	11	0		

Bit Positions 4, 5, 6, and 7 ↓ / Bit Positions 0 and 1 →	00				01				Digit Punches
Bit Positions 2 and 3 →	00	01	10	11	00	01	10	11	
1001		EM							8-1
1010	SMM	CC	SM		¢	!	⑮	:	8-2
1011	VT	CU1	CU2	CU3	.	$	,	#	8-3
1100	FF	IFS		DC4	<	*	%	@	8-4
1101	CR	IGS	ENQ	NAK	(	)	_	'	8-5
1110	SO	IRS	ACK		+	;	>	=	8-6
1111	SI	IUS	BEL	SUB	|	¬	?	"	8-7
Zone Punches	9 12	9 11	9 0	9	12	11	0		

Bit Positions 4, 5, 6, and 7 ↓ / Bit Positions 0 and 1 →	10				11				Digit Punches
Bit Positions 2 and 3 →	00	01	10	11	00	01	10	11	
1010									8-2
1011									8-3
1100									8-4
1101									8-5
1110									8-6
1111									8-7
Zone Punches	12 0	12 11	11 0	12 11 0	9 12 0	9 12 11	9 11 0	9 12 11 0	

① 12-0-9-8-1
② 12-11-9-8-1
③ 11-0-9-8-1
④ 12-11-0-9-8-1
⑤ No Punches
⑥ 12
⑦ 11
⑧ 12-11-0
⑨ 12-0
⑩ 11-0
⑪ 0-8-2
⑫ 0
⑬ 0-1
⑭ 11-0-9-1
⑮ 12-11

USASCII	EBCDIC Equivalents
RS	IRS
ETB	EOB
ESC	PRE
FS	IFS

Because the International Telegraph Alphabet No. 2 and the Figure Protected Code ZSC3 vary from country to country, the BTAM-supplied translation tables RCTW, RCT3, SCTW, and SCT3 may not fit a given installation. Therefore, four macro instructions, TRSLRCTW, TRSLRCT3, TRSLSCTW, and TRSLSCT3 are provided to modify these tables to produce new tables for use with the TRNSLATE macro instruction. These macros both modify the tables and cause them to be assembled into the user program, so it is not necessary to use the ASMTRTAB macro instruction.

TRSLRCTW and TRSLRCT3 Macro Instructions

Name	Operation	Operand
symbol	{TRSLRCTW / TRSLRCT3}	Fx=code,...

symbol
: is the name to be given to the modified table (i.e., the name that will be specified in the TRNSLATE macro instruction). If symbol is omitted, the original name, IECTRCTW or IECTRCT3, is the name of the modified table.

TRSLRCTW
: specifies that table RCTW is to be modified and assembled.

TRSLRCT3
: specifies that table RCT3 is to be modified and assembled.

Fx=code
: specifies what modification is to be made. F stands for figures shift, x represents the number of the code combination to be translated. The permissible values of x are:
<u>For TRSLRCTW:</u> 1, 28 3, 6, 7, 8, 10 through 14, 19, 22, 24, 26, and 32.
<u>For TRSLRCT3:</u> 1, 5, 8, 9, 11, 12, 14, 15, 17 through 20, 22, 24, 26, and 32.

<u>Example</u>: If the transmission code used by a WT terminal is the International Telegraph Alphabet No. 2, combination 6 in figures shift, representing the % character, does not exist in table RCTW. Therefore, you would modify table RCTW by coding

```
TRSLRCTW  F6=6C
```

where 6C is the hexadecimal representation of the % character in EBCDIC.

TRSLSCTW and TRSLSCT3 Macro Instructions

Name	Operation	Operand
symbol	{TRSLSCTW / TRSLSCT3}	Xyy=Fx,...

symbol
: is the name to be given to the modified table (i.e., the table name that will be specified in the TRNSLATE macro instruction. If symbol is omitted, the original name, IECTSCTW or IECTSCT3, is the name of the modified table.

TRSLSCTW
: specifies that table SCTW is to be modified and assembled.

TRSLSCT3
: specifies that table SCT3 is to be modified and assembled.

Xyy=Fx
: specifies what modification is to be made. yy is the hexadecimal representation, in EBCDIC, of the character to be translated. x is the number of the code combination for the character to which yy is to be translated. (F stands for figures shift.) The permissible values of yy are : 2A, 3F, 4A through 50, 5A through 61, 6A through 6F, and 7A through 7F.

<u>Example</u>: If the transmission code used by a WT terminal is the ITA No. 2, and if you wish to translate an EBCDIC % character (hexadecimal 6C in EBCDIC) to an ITA No. 2 % character (combination 6 in figures shift), you would code:

```
TRSLSCTW  X6C=F6
```

Similarly, if you wish to translate an EBCDIC * character (hexadecimal 5C in EBCDIC) to a % character, you would code:

```
TRSLSCTW  X5C=F6
```

And if you wish both the % and * characters to be translated to % characters, you would code:

```
TRSLSCTW  X6C=F6,X5C=F6
```

Note: You can code the same macro several times, each with a different name, to create as many translation tables as needed. This permits several terminals using the same transmission code, but varying character arrangements to operate in the same installation.

Code Combination No.	Elements 12 345	Lettershift Hex Code	Lettershift Character	Figureshift Hex Code	Figureshift Character ITA2	Figureshift Character ZSC3
1	11 000	18	A	38	-	+
2	10 011	13	B	33	?	6
3	01 110	0E	C	2E	:	8
4	10 010	12	D	32	Who are you	
5	10 000	10	E	30	3	-
6	10 110	16	F	36	N/A	4
7	01 011	0B	G	28	N/A	0
8	00 101	05	H	2B	N/A	?
9	01 100	0C	I	2C	8	Bell
10	11 010	1A	J	3A	Bell	2
11	11 110	1E	K	3E	(	(
12	01 001	09	L	29	)	)
13	00 111	07	M	27	.	7
14	00 110	06	N	26	,	,
15	00 011	03	O	23	9	:
16	01 101	0D	P	2D	0	9
17	11 101	1D	Q	3D	1	N/A
18	01 010	0A	R	2A	4	/
19	10 100	14	S	34	'	[illegible]
20	00 001	01	T	21	5	.
21	11 100	1C	U	3C	7	1
22	01 111	0F	V	2F	=	=
23	11 001	19	W	39	2	3
24	10 111	17	X	37	/	N/A
25	10 101	15	Y	35	6	5
26	10 001	11	Z	31	+	N/A
27	00 010	02		22	CR	CR
28	01 000	08		28	LF	LF
29	11 111	1F		3F	LTRS	LTRS
30	11 011	1B		3B	FIGS	FIGS
31	00 100	04		24	Space	Space
32	00 000	00		20	N/A	N/A

Note: N/A = Not assigned
CR = Carriage return
LF = Line feed
LTRS = Letters shift
FIGS = Figures shift

Table 48. World Trade Telegraph Codes ITA2 and ZSC3

Each operand of each BTAM macro instruction can be coded in one or more ways, as indicated in the table to the right. Listed below are the meanings of each of the column headings in the table. The same information is given for each of the user-table-defining macro instructions for the IBM 2715; see the table following the BTAM Macro Instructions table.

Abbreviations Used in Macro Instruction Tables

Abbreviation	Meaning
	You may code the operand as:
Sym	Any symbol valid in the assembler language.
Dec Dig	Decimal digits, within the range shown in the macro instruction description. (The sequence of digits is assembled as a single integer, not as individual digits.)*
Register	Register notation; i.e., a number of a general register, enclosed in parentheses. You must previously have loaded the specified register with the value or address indicated in the operand description. The value or address must be right-adjusted in the register, with all high-order bits set to zero. You may specify registers 2-12 symbolically (CTREG5), or with an absolute expression (5). Registers 0 and 1 can only be specified absolutely: (0), (1).
Char	Any character self-defining term, coded without the framing characters, C' '.
Dec Char	Concatenated decimal digits (each digit is individually assembled in binary format).*
Hex Char	Concatenated hexadecimal digits, coded without the framing characters, X' '.
Code	One of the coded values as given in the individual macro instruction description.
RX-type	Any address that is valid in an RX-type instruction (e.g., LA).
Rel Exp	A relocatable expression (acceptable as an A-type or V-type address constant by the assembler).
Abs Exp	Any absolute expression as defined by the assembler: self-defining terms (decimal, hexadecimal, binary, character), length attributes, absolute symbols, paired relocatable terms in the same control section (CSECT), and arithmetic combinations of absolute terms.

---

*The distinction between Dec Dig and Dec Char may be illustrated by two examples:
19 coded where Dec Dig is specified is assembled as binary 1 0011;
19 coded where Dec Char is specified is assembled as binary 0000 0001 0000 1001.

267 coded where Dec Dig is specified is assembled as binary 1 0000 1011; 267 coded where Dec Char is specified is assembled as binary 0000 0010 0110 0000 0111.

Macro Instruction	Operand	Sym	Dec Dig	Register			RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
				(2-12)	(1)	(0)							
ASMTRTAB	tablename												X
CHGNTRY	listaddr**			X				X					
	dcbaddr***			X				X					
	listype												X
	listposition			X					X				
	numchars**			X					X				
	action												X
CLOSE	dcb							X					
	MF=												X
	listname			X	X		X						
DCB	DSORG=												X
	MACRF=												X
	DDNAME=	X											
	BUFNO=								X				
	BUFL=								X				
	BUFCB=							X					
	EXLST=							X					
	BFTEK=												X
	LERB=							X					
	EROPT=												X
	DEVD=												X
	MODE=												X
	CODE=												X
DFTRMLST	listype												X
	xx											X	
	yy											X	
	dialcount		X										
	dialchars										X		
	numsent		X										
	sentchar											X	
	numcnsent		X										

* see macro description for allowable values
** does not apply to local 3270 display system
*** applies only to local 3270 display system

Table G. BTAM Macro Instruction Format Charts (Part 1 of 6)

Macro Instruction	Operand	Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
DFTRMLST (Continued)	cntrlseq											X	
	tidseq											X	
	numrec		X										
	ridseq											X	
	AN												As Shown
	MD												As Shown
	AD												As Shown
	entrylength		X										
	userlength		X										
	idcount		X										
	idsent											X	
	authsequence											X	
	controlvalue		X										
	userdata							X					
LERB	nlines								X				
	transmct								X				
	datack								X				
	intreq								X				
	notto								X				
LERPRT	dcbaddr	X		X	X								
	rln			X		X			X				
	cid			X				X					
	CLEAR=												X
LOPEN	decbaddr	X		X									
ONLTST	DECB=			X	X		X						
	X=			X					X				
	Y=			X					X				
	DCB=			X			X						
	AREA=			X			X						
	TEXT=			X			X						
	LENGTH=			X					X				
	ENTRY=			X			X						
	RLN=			X					X				
OPEN	dcb							X					
	MF=												X
	listname			X	X		X						

Table G. BTAM Macro Instruction Format Charts (Part 2 of 6)

Macro Instruction	Operand	Sym	Dec Dig	Register			RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
				(2-12)	(1)	(0)							
READ (list form, MF=L)	decbaddr	X											
	optype												X
	dcbaddr							X					
	inoutarea							X					
	inarea							X					
	outarea							X					
	inoutlength								X				
	inlength								X				
	outlength								X				
	entry							X					
	rln								X				
	MF=L												As Shown
READ (Execute form, MF=E)	decbaddr			X	X		X						
	optype												X
	dcbaddr			X			X						
	inoutarea			X			X						'S'
	inarea			X			X						'S'
	outarea			X			X						
	inoutlength			X					X				'S'
	inlength			X					X				'S'
	outlength			X					X				
	entry			X			X						'S'
	rln			X					X				
	MF=E												As Shown
READ (Standard form)	decbaddr	X											
	optype												X
	dcbaddr			X				X					
	inoutarea			X				X					'S'
	inarea			X				X					'S'
	outarea			X				X					
	inoutlength			X					X				'S'
	inlength			X					X				'S'
	outlength			X					X				
	entry			X				X					
	rln			X					X				
RELBUF	dcbaddr			X	X		X						
	bufferaddr			X									

* see macro description for allowable values

Table G. BTAM Macro Instruction Format Charts (Part 3 of 6)

Macro Instruction	Operand	Sym	Dec Dig	Register			RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
				(2-12)	(1)	(0)							
REQBUF	dcbaddr			X	X		X						
	returnreg			X									
	count			X		X			X				
RESETPL	decbaddr			X	X			X					
	POLLING												As Shown
	ANSRING												As Shown
	ATTENT												As Shown
TRNSLATE	dcbaddr			X			X						
	tablename			X			X						
	area			X			X						
	length			X		X			X				'S'
TRSLRCTW	Fnn=												X
TRSLRCT3	Fnn=												X
TRSLSCTW	Xyy=												X
TRSLSCT3	Xyy=												X
TWAIT	returnreg			X									
	ECBLIST=			X			X						
WAIT	count			X		X	X		X				
	ECB=			X	X		X						
	ECBLIST=			X	X								
WRITE (List form, MF=L)	decbaddr	X											
	optype												X
	dcbaddr							X					
	inoutarea							X					
	inarea							X					
	outarea							X					
	inoutlength								X				
	inlength								X				
	outlength								X				
	entry							X					
	rln								X				
	MF=L												As Shown

* see macro description for allowable values

Table G. BTAM Macro Instruction Format Charts (Part 4 of 6)

Macro Instruction	Operand	Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
WRITE (Execute form, MF=E)	decbaddr			X	X		X						
	optype												X
	dcbaddr			X			X						
	inoutarea			X			X						
	inarea			X			X						'S'
	outarea			X			X						
	inoutlength			X					X				'S'
	inlength			X					X				'S'
	outlength			X					X				
	entry			X			X						
	rln			X					X				
	MF=E												As Shown
WRITE (Standard form)	decbaddr	X											
	optype												X
	dcbaddr	X						X					
	inoutarea	X						X					
	inarea	X						X					'S'
	outarea	X						X					
	inoutlength	X							X				'S'
	inlength	X							X				'S'
	outlength	X							X				
	entry	X						X					
	rln	X							X				

* see macro description for allowable values

Table G. BTAM Macro Instruction Format Charts (Part 5 of 6)

2715 User-Table Macro Instructions

Macro Instruction	Operand	Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code *
As	ID=								X				
	ASGROUP=	X											
	DEGROUP= tgroupname deunumber	 X 							 X				
ASCTR	ID=								X				
	HIGHCTR=								X				
	ROUTE=												X
	LOG												As shown
	ASLOG												As shown
	EXTALRM												As shown
	NEXTAS								X				
ASLIST	device												X
	NORM=								X				
	LENGTH= data length gdlight2								 X X				
	DIGIT= entrypos compvalue gdlight3								 X X X				
	ENTRY=												X
	MSG=									X			
	INQDISP=								X				
	MODULUS= entrypos data length gdlight4								 X X X				
	SELTRAN=												X

* see macro description for allowable values

Table G: BTAM Macro Instruction Format Charts (6A of 6)

2715 User-Table Macro Instructions

Macro Instruction	Operand	Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	RX-type	Rel Exp	Abs Exp	Char	Dex Char	Hex Char	Code *
CONFIGUR	CORE=												X
	PC=												X
	GDU=												X
	FUNCERR=								X				
	ENDERR=								X				
	MONERR=								X				
	GETID=								X				
	STORID=								X				
	IDCOUNT=								X				
	INQDISP=												X
CTRGROUP	ctrno								X				
	sro								X				
	cttest								X				
	ID=								X				
	SROENAB=												X
	CTINIT=												X
CTRLIST	DEVCOD=												X
	CTRADR=												X
	CTRRD=												X
	CTTEST=												X
	CTROP=												X
	MSG=									X			
CTRSCHED	sched								X				

* see macro description for allowable values

Table G: BTAM Macro Instruction Format Charts (6B of 6)

2715 User-Table Macro Instructions

Macro Instruction	Operand	Sym	Dec Dig	Register			RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code *
				(2-12)	(1)	(0)							
DEULIST	LENGTH=		X										
	DIGIT entrypos compvalue								 X X				
	MSG=									X			
	MODULUS= entrypos data length								 X X				
	DIGIT2= entrypos compvalue								 X X				
DISPGUID	DISPMSG=									X			
	SUPPRES=												X
GDUAS	ID=								X				
	GDUNUMB=								X				
GDULIST	PARAMNO=								X				
	NORGUID=								X				
	DISPMSG=	X											
	IDENT=								X				
	MSG=									X			
	ENTRY=												X
GDUTRANS	TRCODE=								X				
	TRLIST=	X											
PARAMNUM	PLN=								X				
	PARMLST=	X											

* see macro description for allowable values

Table G: BTAM Macro Instruction Format Charts (6C of 6)

2715 User-Table Macro Instructions

Macro Instruction	Operand	Sym	Dec Dig	Register			RX-type	Rel Exp	Abs Exp	Char	Dex Char	Hex Char	Code *
				(2-12)	(1)	(0)							
PARMLIST	CKLNGTH=												
	data length								X				
	gdlight								X				
	CKMONKY=												X
	CKMOD11=												
	data length								X				
	entrypos								X				
	gdlight								X				
	CKRANGE=												
	firstpos								X				
	lastpos								X				
	compvalue								X				
	LOWGUID=								X				
	HIGUID=								X				
	RNGETST=												X
	CKMOD10=												
	data length								X				
	entrypos								X				
	gdlight								X				
	CKOR=												
	data pos								X				
	checkchar											X	
	ORGUID=								X				
	CKAND=												
	startpos								X				
	endpos								X				
	checkchar											X	
	ANDGUID=								X				
	CKNONUM=												
	startpos								X				
	endpos								X				
	gdlight								X				
	CKNUM=												
	startpos								X				
	endpos								X				
	gdlight								X				
	TRANSL=												X
	IDENT=												X

* see macro description for allowable values

Table G: BTAM Macro Instruction Format Charts (6D of 6)

2715 User-Table Macro Instructions

Macro Instruction	Operand	Sym	Dec Dig	Register			RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code *
				(2-12)	(1)	(0)							
STEND	no operands												
TGROUP	TCn= tcode E	X											As shown
TRANSLAT	TRANSCH=											X	
	TRANTXT=									X			
TRLIST	TRID=								X				
	ROUTE=												X
	LOG												As shown
	NULL												As shown
	asaddr								X				
	TEXT=												X
	INQDISP=												X
	DEMOD10=												X
	DEMOD11=												X
	GDU=												X

* see macro description for allowable values

Table G: BTAM Macro Instruction Format Charts (6E of 6)

This chart shows the character set and bit patterns for the Extended Binary Coded Decimal Interchange Code (EBCDIC), and the character sets and transmission code bit patterns for each of the remote station types supported by BTAM.

The chart may be used to determine the bit patterns, as contained in main storage bytes, for each of the various characters sent or received by a specific type of station, and to determine the relationships, as established by the arrangement of the IBM-provided translation tables, among the character sets for the various types.

For convenience in referring to particular chart locations, the chart's columns and rows are given reference numbers. Combined, these numbers enable reference to a particular chart location; e.g., location 21/17, the intersection of row 21 and column 17, contains NL.

Arrangement of Chart

The chart contains a group of three columns for the EBCDIC character set and a group for each of the various terminal character sets. Within the EBCDIC group, column 3 contains the 256 bit patterns comprising the code. For those bit patterns to which characters are currently assigned, the characters appear in column 1 (graphics) and column 2 (line controls and device controls). All currently assigned characters are shown, regardless of whether they are in the character sets of any of the types of remote stations represented in the remainder of the chart.

Each of the remaining groups (columns 4 through 33) contains the characters comprising the character set of a specific station type, along with the transmission code bit patterns. Column 34 repeats the EBCDIC code presented in column 3, for ease of reference.

In the EBCDIC group, the bit patterns and characters are arranged in collating sequence from hexadecimal 00 to hexadecimal FF. In the remainder of the chart, the locations of bit patterns and characters are determined by the arrangement of the translation tables.

Character Sets

This chart shows only the characters comprising the commonly used character set options. The options represented in the chart are:

Terminal	Option
IBM 1030	Standard and "H" options
IBM 1050	System/360 option
IBM 1060	Standard option
IBM 2260	Standard option
IBM 2740	System/360 option
AT&T 83B3, WU 115A	"A" and "C" options
WU TWX	Standard option

IBM 1030 graphics and AT&T 83B3/WU 115A graphics that differ for the respective options are indicated in the chart by S and H, and A and C, respectively. Graphics not so marked are the same in both options.

Transmission Codes

The notations in the code columns of the chart for the various types of stations represent the System/360 byte bit pattern equivalents of the applicable transmission codes. The applicable transmission codes are:

Terminal	Code
IBM 1030	Perforated tape and transmission code.
IBM 1050	Perforated tape and transmission code
IBM 1060	Perforated tape and transmission code
IBM 2260	IBM 2260 transmission code
IBM 2740	Perforated tape and transmission code (BCD code)
AT&T 83B3	5-level Baudot code

WU 115A	5-level Baudot code
WU TWX	8-level TWX code

Representation of Characters and Bit Patterns

Appearance of a character and its associated bit pattern in a character set signifies that the appropriate IBM-provided translation tables effect either incoming translation (i.e., translation of that character to the corresponding EBCDIC character), or outgoing translation (i.e., translation of the corresponding EBCDIC character to that character), or both. How the bit pattern appears indicates which of these cases applies:

1. Where the hexadecimal representation of the bit pattern appears in brackets, only incoming translation is performed.

2. Where the bit pattern is enclosed in parentheses, only outgoing translation is performed.

3. Where the bit pattern is not enclosed by brackets or parentheses, both incoming and outgoing translation are performed.

Because each unique bit pattern for a terminal character can be represented only once in an "incoming" translation table, the character associated with the bit pattern can be translated to only one EBCDIC character. The converse is not true, however; any one transmission code bit pattern can be placed any number of times within an "outgoing" table. Therefore, any number of EBCDIC characters can be translated to the terminal character represented by that bit pattern.

Appearance of two bit patterns opposite a single character signifies that the character has both an upper-case (or figures shift) and a lower-case (or letters shift) bit pattern, and that both forms of the character are translated to the same EBCDIC character. (Exception: In the code column for TWX terminals, where two bit patterns appear, the left-hand one is the even-parity pattern, and the right-hand one is the non-parity pattern.)

Example: The bit pattern of the NL character appears in location 21/9. Both the lower- and upper-case bit patterns of this character are translated to the EBCDIC NL character when they appear in an incoming message. When an EBCDIC NL character appears in an outgoing message, BTAM translates it to the lower-case form of the NL character.

Where more than one EBCDIC character requires translation to the same character in a terminal character set, the terminal character appears an equivalent number of times in the column (e.g., locations 0/23, 6/23, 7/23, 23/23, and 50/23 all contain the LTRS character).

Where a character appears in both the graphics and the controls columns for a terminal type, its function depends on whether it is sent when the line is in control mode or in text mode. Depending on the type of terminal and the mode, the character may perform a control function, print as a graphic, or both. For details, see the reference manuals for the various terminal types.

Nonequivalent Characters

Designing the system to accomodate terminal types having different character sets and control functions has resulted in several instances where dissimilar characters have been "equated" in translation tables. This accounts for the appearance in certain rows of this chart of non-equivalent characters, for example, in rows 3, 38, and 50.

In other instances, the same or similar functions have different names among the various terminal types; for example, HT and Tab in row 5 are equivalent, as are DEL and Rubout in row 7. In a few cases, terminals using the same transmission code have different meanings assigned to the identical bit pattern; for example, bit pattern 79 in the transmission code has the meaning PF for an IBM 1050, and Subtract for an IBM 1060.

Substitutions

Where blank positions appear in the terminal character set portion of the chart, there is no equivalent character for the EBCDIC character or bit pattern at the left of the chart. Where these blanks appear, the SUB character is to be assumed (they were omitted to make the chart more readable). That is, in each translation table that handles incoming messages, each position representing an invalid transmission code bit pattern (i.e., one not used by a character in the terminal's character set) contains the EBCDIC code (3F) for the SUB character. In each translation table that handles outgoing messages, the transmission code bit pattern for a substitute graphic is contained in each of the following positions:

- Each position that represents an invalid EBCDIC bit pattern (a pattern to which no EBCDIC character has been assigned).
- Each position that represents a bit pattern for a character having no equivalent in the destination terminal's character set.

For the IBM 1050, 2260, and 2740, and the AT&T 83B3 and WU 115A, this substitute character is a colon (:). For the IBM 1030 and 1060, and the WU TWX, it is a slash (/).

General Notes

1. Standard abbreviations are used to represent the control characters. The full names of the characters are given in a following section entitled Control Characters. For descriptions of these characters, see the reference manuals for the various terminals.

2. Where a "circle" character (Ⓑ, Ⓓ, etc.) appears in parentheses adjacent to a control character, it is an alternate name for that control character.

3. Notes pertaining to specific characters or bit patterns are indicated by superscript numerals next to the character or bit pattern. The notes appear below, and indicate the chart locations to which they apply.

4. Most of the characters in the "S" and "H" character set options (1030) and in the "A" and "C" character set options (83B3, 115A) are identical. Where they differ between the options, the translation tables "favor" the "S" option and the "A" option, as illustrated in the chart. If messages from an "H" option 1030 are sent only to another "H" option 1030, the translation table may be used as is, and similarly, for the 83B3/115A, with respect to the "C" option. If messages from terminals with the "H" or "C" option are to be exchanged with other terminal types, you may wish to modify the tables.

5. Some TWX terminals send even-parity transmission code bit patterns; others send non-parity bit patterns. All bit patterns sent by non-parity machines have a "1" in the low-order bit position (i.e., the position that serves as the parity bit in even-parity machines). The RCT2 translation table translates either a non-parity or an even-parity bit pattern to the EBCDIC bit pattern for the corresponding character. The SCT2 translation table always sends even parity.

Notes:

[1]Left bracket translates to EBCDIC hex 79; no EBCDIC character has been assigned to this bit pattern (location 121/3, 121/25).

[2]No graphic prints in the "A" character set option (location 90/22).

[3]Backslash translates to EBCDIC hex E1; no EBCDIC character has been assigned to this bit pattern (locations 225/3, 225/25).

[4]IBM 1031 sends the numeric 0 as a hex 20; 1033 receives the numeric 0 as a hex 15 (location 240/4).

[5]Right bracket translates to EBCDIC hex 49; no EBCDIC character has been assigned to this bit pattern (locations 73/3, 73/25).

Control Characters

ACK	Positive Acknowledgment
Ⓑ	End-of-block (same as EOB)
BEL	Bell
BS	Backspace
BYP	Bypass
Ⓒ	End-of-transmission (same as EOT)
CAN	Cancel
CC	Cursor control
CR	Carriage (carrier) return
Ⓓ	Machine end-of-address (same as EOA)
DC1 DC2 DC4	Device controls
DEL	Delete

DLE	Data link escape	NUL	Null
DS	Digit select	PF	Punch off
EM	End of medium	PN	Punch on
ENQ	Enquiry	PRE	Prefix
EOA	End-of-address	PZ	Plus zero
EOB	End-of-block	RES	Restore
EOC	End of card	RM	Record mark
EOFC	End of first card	RS	Reader stop
EOM	End-of-message	S	Start-of-address
EOT	End-of-transmission	SI	Shift in
ETB	End-transmission-block	SM	Set mode
ETX	End-of-text	SO	Shift out
FF	Forms feed	SOH	Start-of-header
FIGS	Figures shift	SMM	Start-manual-message
FS	Field separator	SOS	Start-of-significance
HT	Horizontal tabulate	SP	Space
IFS	Interchange file separator	STX	Start-of-text
IGS	Interchange group separator	SUB	Substitute
IL	Idle	SYN	Synchronous idle
IRS	Interchange record separator	Tab	Tabulate (horizontal)
IUS	Interchange unit separator	TM	Tape mark
LC	Lower-case shift	TpAuxOff	Tape auxiliary off
LF	Line feed	TpAuxOn	Tape auxiliary on
LF-CR	Line feed-carriage return	UC	Upper-case shift
LTRS	Letters shift	VT	Vertical Tabulate
MZ	Minus zero	WRU	'Who Are You?'
N	Negative response to polling, addressing, or LRC/VRC	X-Off	Transmitter off
NAK	Negative acknowledgment	X-On	Transmitter on
NL	New line	Y	Positive response to polling, addressing, or LRC/VRC

	EBCDIC			IBM 1030			IBM 1050			IBM 1060			IBM 2260					
													2260			1053		
	Character		Code	Character		Code	Character		Code	Character		Code	Character		Code	Character		Code
	Graphic	Control	(Hex)	Graphic	Control	(Hex)	Graphic	Control	(Hex)	Graphic	Control	(Hex)	Graphic	Control	(Hex)	Graphic	Control	(Hex)
Ref.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0		NUL	00		Pad	(DF)		IL	(5E)		IL	(5E)						
1		SOH	01		Ⓢ	(37)								SOH	01		SOH	(01)
2		STX	02	s# H=	EOA(Ⓓ)	(16)	#	EOA(Ⓓ)	(16)	#	EOA(Ⓓ)	(16)		STX	02		STX	(02)
3		ETX	03		EOB(Ⓑ)	(3D)		EOB(Ⓑ)	(3D)		EOB(Ⓑ)	(3D)		ETX	03		ETX	(03)
4		PF	04					PF	79 [F9]		Subtr	(79)						
5		HT	05		HT	(7A)		Tab	7A [FA]		Tab	(7A)						
6		LC	06		Pad	(DF)		Dwnshft	7C [FC]		IL	(5E)						
7		DEL	07		EOC DEL	7F		DEL	7F [FF]		DEL	7F						
8			08															
9			09															
10		SMM	0A										▷	Start MI	[BD]	¢		(BD)
11		VT	0B															
12		FF	0C															
13		CR	0D		LF-CR	(5B)		NL	(5B)		CR	(5B)		NL	(0A)		NL	(0A)
14		SO	0E					BYP	(38)									
15		SI	0F					RES	(58)									
16		DLE	10															
17		DC1	11															
18		DC2	12															
19		TM	13															
20		RES	14					RES	58 [DB]	★		(58)						
21		NL	15		LF-CR	(5B)		NL	5B [DB]		CR	(5B)	◿	NL	0A		NL	(0A)
22		BS	16					BS	5D [DD]									
23		IL	17		Pad	(DF)		IL	5E [DE]		IL	(5E)						
24		CAN	18											CAN	[18]			
25		EM	19															
26		CC	1A										▯	Check	42	"		(42)
27		CU1	1B															
28		IFS	1C															
29		IGS	1D															
30		IRS	1E															
31		IUS	1F															
32		DS	20															
33		SOS	21															
34		FS	22															
35			23															
36		BYP	24					BYP	38 [B8]									
37		LF	25		LF	(3B)		LF	3B [BB]		LF	(3B)						
38		ETB (EOB)	26		EOB(Ⓑ)	3D		EOB	3D [BD]		EOB(Ⓑ)	3D		ETX	(03)		ETX	(03)
39		ESC (PRE)	27					PRE	3E [BE]									
40			28															
41			29															
42		SM	2A															
43		CU2	2B															
44			2C															
45		ENQ	2D															
46		ACK	2E											ACK	06		ACK	06
47		BEL	2F															
48			30															
49			31															
50		SYN	32		Pad	(DF)		IL	(5E)		IL	(5E)						
51			33															
52		PN	34					PN	19 [99]									
53		RS	35					RS	1A [9A]									
54		UC	36					Upshift	1C [9C]									
55		EOT	37		EOT(©)	1F		EOT(©)	1F		EOT(©)	(1F)		EOT (©)	04		EOT(©)	04
56			38															
57			39															
58			3A															
59		CU3	3B															
60		DC4	3C															
61		NAK	3D											NAK	15		NAK	[15]
62			3E															
63		SUB	3F	/		(23)	:		(88)	/		(23)	:		(5A)	:		(5A)
64		SP	40		SP	01		SP	01 [81]		SP	01		SP	40		SP	(40)
65			41															
66			42															
67			43															
68			44															
69			45															
70			46															
71			47															

IBM 2740			AT&T 83 B3 W U-115A			WU TWX			WTTA (1TA2)			WTTA (ZSC3)			EBCDIC	
Character		Code	Character		Code	Character		Code (Hex)	Character		Code	Character		Code	Code	
Graphic	Control	(Hex)	Graphic	Control	(Hex)	Graphic	Control	Even Non	Graphic	Control	(Hex)	Graphic	Control	(Hex)	(Hex)	
19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	Ref.
	IL	(5E)		LTRS	(1F)		Rubout	(FF) (FF)		LTRS	(1F)		LTRS	(1F)	00	0
	Ⓢ	(37)													01	1
#	EOA(Ⓓ)	(16)													02	2
	EOB(Ⓑ)	(3D)		CR	(02)		CR	(B1) (B1)		CR	(02)		CR	(02)	03	3
							TpAuxOff	28 29							04	4
	HT	7A [FA]					HT	90 91							05	5
	Dwnshft	7C [FC]		LTRS	1F [3F]					LTRS	(1F)		LTRS	(1F)	06	6
	DEL	7F [FF]		LTRS	(1F)		Rubout	FF FF		LTRS	(1F)		LTRS	(1F)	07	7
															08	8
															09	9
															0A	10
							VT	D1 D1							0B	11
							FF	30 31							0C	12
	NL	(5B)		CR	(02)		CR	(B1) (B1)		CR	(02)		CR	(02)	0D	13
							SO	71 71							0E	14
							SI	F0 F1							0F	15
															10	16
							X-On	88 89							11	17
															12	18
															13	19
															14	20
	NL	5B [DB]		LF	(08) [28]		LF	50 [51]		CR	02 [22]		CR LF	02 [22]	15	21
	BS	5D [DD]													16	22
	IL	5E [DE]		LTRS	(1F)		Rubout	(FF)		LTRS	(1F)		LTRS	(1F)	17	23
															18	24
															19	25
															1A	26
															1B	27
															1C	28
															1D	29
															1E	30
															1F	31
															20	32
															21	33
															22	34
															23	35
															24	36
	LF	3B [BB]		LF	(08)		LF	(50) (51)		LF	08 [28]		LF	08 [28]	25	37
	EOB	3D [BD]		CR	[02] [22]		CR	B1		LF	(08)		LF	(08)	26	38
															27	39
															28	40
															29	41
										Bell	3A		Bell	2C	2A	42
															2B	43
															2C	44
							WRU	A0 A1							2D	45
															2E	46
			C,	$_A$Bell	3A		Bell	E1 E1							2F	47
															30	48
															31	49
	IL	(5E)		LTRS	(1F)		Rubout	(FF) (FF)		LTRS	(1F)		LTRS	(1F)	32	50
															33	51
							TpAuxOn	48 49							34	52
															35	53
	Upshft	1C [9C]		FIGS	1B [3B]					FIGS	1B [3B]		FIGS	1B [3B]	36	54
	EOT(©)	1F	#		(25)		EOT	21 21		LTRS	1F		LTRS	(1F)	37	55
															38	56
															39	57
															3A	58
															3B	59
							X-Off	C9 C9							3C	60
															3D	61
															3E	62
:		(88)	/		(37)	:		(5C) (5D)	/		(37)	/		(2A)	3F	63
	SP	01 [81]		SP	04 [24]		SP	05 05		SP	04 [24]		SP	04 [24]	40	64
															41	65
															42	66
															43	67
															44	68
															45	69
															46	70
															47	71

Ref.	EBCDIC			IBM 1030			IBM 1050			IBM 1060		
	Character		Code	Character		Code	Character		Code	Character		Code
	Graphic	Control	(Hex)	Graphic	Control	(Hex)	Graphic	Control	(Hex)	Graphic	Control	(Hex)
	1	2	3	4	5	6	7	8	9	10	11	12
72			48									
73			49 [5]									
74	¢		4A				¢		A0			
75	.		4B		Ⓨ EOFC	76	.	Ⓨ	76	.	Ⓨ	(76)
76	<		4C				<		84			
77	(		4D				(		93			
78	+		4E				+		E1			
79	|		4F				|		B7			
80	&		50	& H+		61	&		61	+		(61)
81			51									
82			52									
83			53									
84			54									
85			55									
86			56									
87			57									
88			58									
89			59									
90	!		5A				!		D7			
91	$		5B	$		57	$		57	$		(57)
92	*		5C				*		90			
93	)		5D				)		95			
94	;		5E				;		87			
95	¬		5F				¬	Ⓨ	F6			
96	-		60	-	Ⓝ	40	-	Ⓝ	40	-	Ⓝ	(40)
97	/		61	/		23	/		23	/		23
98			62									
99			63									
100			64									
101			65									
102			66									
103			67									
104			68									
105			69									
106		EOM	6A									
107	,		6B		Ⓢ	37	,		37	,		(37)
108	%		6C				%		8B			
109	_		6D				_	Ⓝ	C0			
110	>		6E				>		8E			
111	?		6F				?		A3			
112			70									
113			71									
114			72									
115			73									
116			74									
117			75									
118			76									
119			77									
120			78									
121			79 [3]									
122	:		7A				:		88			
123	#	EOA	7B	S# H=	EOA(Ⓓ)	16	#	EOA(Ⓓ)	16	#	EOA(Ⓓ)	16
124	@		7C	S@ H'		(20)	@		20		Add	(20)
125	'		7D				'		8D			
126	=		7E				=		82			
127	"		7F				"	EOA (Ⓓ)	96			
128			80									
129	a		81	A		(62)	a		62	A		(62)
130	b		82	B		(64)	b		64	B		(64)
131	c		83	C		(67)	c		67	C		(67)
132	d		84	D		(68)	d		68	D		(68)
133	e		85	E		(6B)	e		6B	E		(6B)
134	f		86	F		(6D)	f		6D	F		(6D)
135	g		87	G		(6E)	g		6E	G		(6E)
136	h		88	H		(70)	h		70	H		(70)
137	i		89	I		(73)	i		73	I		(73)
138			8A									
139			8B									
140			8C									
141			8D									
142			8E									
143			8F									

Ref.	IBM 2260: 2260			IBM 2260: 1053			IBM 2740			AT&T 83B3 W U 115A		
	Character		Code	Character		Code	Character		Code	Character		Code
	Graphic	Control	(Hex)	Graphic	Control	(Hex)	Graphic	Control	(Hex)	Graphic	Control	(Hex)
	13	14	15	16	17	18	19	20	21	22	23	24
72												
73												
74							¢		A0			
75	.		4E	.		(4E)	.	Ⓨ	76			27
76	<		5C	<		(5C)	<		84			
77	(		48	(		(48)	(		93	A(C1/2		3E
78	+		4B	+		(4B)	+		E1			
79	|		FE	|		(FE)	|		B7			
80	&		46	&		(46)	&		61	&		2B
81												
82												
83												
84												
85												
86												
87												
88												
89												
90							!		D7	A^{2} C1/4		36
91	$		44	$		(44)	$		57	$		32
92	*		4A	*		(4A)	*		90			
93	)		49	)		(49)	)		95	A) C3/4		29
94	;		5B	;		(5B)	;		87	A; C3/8		2F
95	¬		FC	¬		(FC)	¬	Ⓨ	F6			
96	-		4D	-		(4D)	-	Ⓝ	40	-		38
97	/		4F	/		(4F)	/		23	/		37
98												
99												
100												
101												
102												
103												
104												
105												
106	▭	EOM	41	!		(41)						
107	,		4C	,		(4C)	,	Ⓢ	37	A, C7/8		26
108	%		45	%		(45)	%		8B			
109	_		BF	_		(BF)	_	Ⓝ	C0			
110	>		5E	>		(5E)	>		8E			
111	?		5F	?		(5F)	?		A3	A? C5/8		33
112												
113												
114												
115												
116												
117												
118												
119												
120												
121												
122	:		5A	:		(5A)	:		88	A: C1/8		2E
123	#		43	#		(43)	#	EOA (Ⓓ)	16		CR	(02)
124	@		E0	@		(E0)	@	EOA (Ⓓ)	20			
125	'		47	'		(47)	'		8D	A'	C Bell	34
126	=		5D	=		(5D)	=		82			
127							"		96	"		31
128												
129	A		(A1)	A		(A1)	a		62	A		(18)
130	B		(A2)	B		(A2)	b		64	B		(13)
131	C		(A3)	C		(A3)	c		67	C		(0E)
132	D		(A4)	D		(A4)	d		68	D		(12)
133	E		(A5)	E		(A5)	e		6B	E		(10)
134	F		(A6)	F		(A6)	f		6D	F		(16)
135	G		(A7)	G		(A7)	g		6E	G		(0B)
136	H		(A8)	H		(A8)	h		70	H		(05)
137	I		(A9)	I		(A9)	i		73	I		(0C)
138												
139												
140												
141												
142												
143												

Ref.	WU TWX				WTTA (ITA2)			WTTA (ZSC3)			EBCDIC	Ref.
	Character		Code (Hex)		Character		Code	Character		Code	Code	
	Graphic	Control	Even	Non	Graphic	Control	(Hex)	Graphic	Control	(Hex)	(Hex)	
	25	26	27		28	29	30	31	32	33	34	
72											48	72
73	] [5]		BB	BB							49	73
74											4A	74
75	.		74	75	.		27	.		21	4B	75
76	<		3C	3D							4C	76
77	(		14	15	(		3E	(		3E	4D	77
78	+		D4	D5	+		31	+		38	4E	78
79	↑		7B	7B							4F	79
80	&		65	65							50	80
81											51	81
82											52	82
83											53	83
84											54	84
85											55	85
86											56	86
87											57	87
88											58	88
89											59	89
90	!		84	85							5A	90
91	$		24	25							5B	91
92	*		55	55							5C	92
93	)		95	95	)		29	)		29	5D	93
94	;		DD	DD						2F	5E	94
95											5F	95
96	-		B4	B5	-		38	-		30	60	96
97	/		F5	F5	/		37	/		2A	61	97
98											62	98
99											63	99
100											64	100
101											65	101
102											66	102
103											67	103
104											68	104
105											69	105
106											6A	106
107	,		35	35	,		26	,		26	6B	107
108	%		A5	A5							6C	108
109	←		FA	FB							6D	109
110	>		7D	7D							6E	110
111	?		FC	FD	?		33	?		25	6F	111
112											70	112
113											71	113
114											72	114
115											73	115
116											74	116
117											75	117
118											76	118
119											77	119
120											78	120
121	[[1]		DB	DB							79	121
122	:		5C	5D	:		2E	'		23	7A	122
123	#		C5	C5							7B	123
124	@		03	03							7C	124
125	'		E4	E5	,		34	,		34	7D	125
126	=		BD	BD	=		2F	=		2F	7E	126
127	"		44	45							7F	127
128											80	128
129	A		(82)	(83)	A		(18)	A		(18)	81	129
130	B		(42)	(43)	B		(13)	B		(13)	82	130
131	C		(C3)	(C3)	C		(0E)	C		(0E)	83	131
132	D		(22)	(23)	D		(12)	D		(12)	84	132
133	E		(A3)	(A3)	E		(10)	E		(10)	85	133
134	F		(63)	(63)	F		(16)	F		(16)	86	134
135	G		(E2)	(E3)	G		(06)	G		(08)	87	135
136	H		(12)	(13)	H		(05)	H		(05)	88	136
137	I		(93)	(93)	I		(0C)	I		(0C)	89	137
138											8A	138
139											8B	139
140											8C	140
141											8D	141
142											8E	142
143											8F	143

IC		IBM 1030			IBM 1050			IBM 1060			IBM 2260: 2260			IBM 2260: 1053			IBM 2740			IBM 7770/7772			AT&T 83B3 W U 115A			WU TWX			WTTA (ITA2)			WTTA (ZSC3)			EBCDIC	
	Code	Character		Code	Character		Code	Character		Code	Character		Code	Character		Code	Character		Code	Character		Code	Character		Code	Character		Code (Hex)	Character		Code (HEX)	Character		Code	Code	
ntrol	(Hex)	Graphic	Control	(Hex)	Graphic	Control	(Hex)	Graphic	Control	(Hex)	Graphic	Control	(Hex)	Graphic	Control	(Hex)	Graphic	Control	(Hex)	Graphic	Control	(Hex)	Graphic	Control	(Hex)	Graphic	Control	Even Non	Graphic	Control	Even Non	Graphic	Control	(Hex)	(Hex)	
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	Ref.
	90																																		90	144
	91	J		(43)	j		43	J		(43)	J		(AA)	J		(AA)	j		43				J		(1A)	J		(53)	J		(1A)	J		(1A)	91	145
	92	K		(45)	k		45	K		(45)	K		(AB)	K		(AB)	k		45				K		(1E)	K		(D2)	K		(1E)	K		(1E)	92	146
	93	L		(46)	l		46	L		(46)	L		(AC)	L		(AC)	l		46				L		(09)	L		(33)	L		(09)	L		(09)	93	147
	94	M		(49)	m		49	M		(49)	M		(AD)	M		(AD)	m		49				M		(07)	M		(B2)	M		(07)	M		(07)	94	148
	95	N		(4A)	n		4A	N		(4A)	N		(AE)	N		(AE)	n		4A				N		(06)	N		(72)	N		(06)	N		(06)	95	149
	96	O		(4C)	o		4C	O		(4C)	O		(AF)	O		(AF)	o		4C				O		(03)	O		(F3)	O		(03)	O		(03)	96	150
	97	P		(4F)	p		4F	P		(4F)	P		(B0)	P		(B0)	p		4F				P		(0D)	P		(0A)	P		(0D)	P		(0D)	97	151
	98	Q		(51)	q		51	Q		(51)	Q		(B1)	Q		(B1)	q		51				Q		(1D)	Q		(8B)	Q		(1D)	Q		(1D)	98	152
	99	R		(52)	r		52	R		(52)	R		(B2)	R		(B2)	r		52				R		(0A)	R		(4B)	R		(0A)	R		(0A)	99	153
	9A																																		9A	154
	9B																																		9B	155
	9C																																		9C	156
	9D																																		9D	157
	9E																																		9E	158
	9F																																		9F	159
	A0																																		A0	160
	A1																																		A1	161
	A2	S		(25)	s		25	S		(25)	S		(B3)	S		(B3)	s		25				S		(14)	S		(CA)	S		(14)	S		(14)	A2	162
	A3	T		(26)	t		26	T		(26)	T		(B4)	T		(B4)	t		26				T		(01)	T		(2B)	T		(01)	T		(01)	A3	163
	A4	U		(29)	u		29	U		(29)	U		(B5)	U		(B5)	u		29				U		(1C)	U		(AA)	U		(1C)	U		(1C)	A4	164
	A5	V		(2A)	v		2A	V		(2A)	V		(B6)	V		(B6)	v		2A				V		(0F)	V		(6A)	V		(0F)	V		(0F)	A5	165
	A6	W		(2C)	w		2C	W		(2C)	W		(B7)	W		(B7)	w		2C				W		(19)	W		(EB)	W		(19)	W		(19)	A6	166
	A7	X		(2F)	x		2F	X		(2F)	X		(B8)	X		(B8)	x		2F				X		(17)	X		(1B)	X		(17)	X		(17)	A7	167
	A8	Y		(31)	y		31	Y		(31)	Y		(B9)	Y		(B9)	y		31				Y		(15)	Y		(9A)	Y		(15)	Y		(15)	A8	168
	A9	Z		(32)	z		32	Z		(32)	Z		(BA)	Z		(BA)	z		32				Z		(11)	Z		(5A)	Z		(11)	Z		(11)	A9	169
	AA																																		AA	170
	AB																																		AB	171
	AC																																		AC	172
	AD																																		AD	173
	AE																																		AE	174
	AF																																		AF	175
	B0																																		B0	176
	B1																																		B1	177
	B2																																		B2	178
	B3																																		B3	179
	B4																																		B4	180
	B5																																		B5	181
	B6																																		B6	182
	B7																																		B7	183
	B8																																		B8	184
	B9																																		B9	185
	BA																																		BA	186
	BB																																		BB	187
	BC																																		BC	188
	BD																																		BD	189
	BE																																		BE	190
	BF																																		BF	191
PZ	C0					PZ	75		Restore	(75)																									C0	192
	C1	A		62	A		E2	A		62	A		A1	A		(A1)	A		E2	A		[C9]	A		18	A		82 [83]	A		18	A		18	C1	193
	C2	B		64	B		E4	B		64	B		A2	B		(A2)	B		E4	B		[D1]	B		13	B		42 [43]	B		13	B		13	C2	194
	C3	C		67	C		E7	C		(67)	C		A3	C		(A3)	C		E7	C		[D9]	C		0E	C		C3	C		0E	C		0E	C3	195
	C4	D		68	D		E8	D		(68)	D		A4	D		(A4)	D		E8	D		[CA]	D		12	D		22 [23]	D		12	D		12	C4	196
	C5	E		6B	E		EB	E		6B	E		A5	E		(A5)	E		EB	E		[D2]	E		10	E		A3	E		10	E		10	C5	197
	C6	F		6D	F		ED	F		6D	F		A6	F		(A6)	F		ED	F		[DA]	F		16	F		63	F		16	F		16	C6	198
	C7	G		6E	G		EE	G		(6E)	G		A7	G		(A7)	G		EE	G		[CB]	G		0B	G		E2 [E3]	G		0B	G		0B	C7	199
	C8	H		70	H		F0	H		(70)	H		A8	H		(A8)	H		F0	H		[D3]	H		05	H		12 [13]	H		05	H		05	C8	200
	C9	I		73	I		F3	I		73	I		A9	I		(A9)	I		F3	I		[DB]	I		0C	I		93	I		0C	I		0C	C9	201
	CA																																		CA	202
	CB																																		CB	203
	CC																																		CC	204
	CD																																		CD	205
	CE																																		CE	206
	CF																																		CF	207
MZ	D0					MZ	54		Message	(54)																									D0	208
	D1	J		43	J		C3	J		43	J		AA	J		(AA)	J		C3	J		[89]	J		1A	J		53	J		1A	J		1A	D1	209
	D2	K		45	K		C5	K		45	K		AB	K		(AB)	K		C5	K		[91]	K		1E	K		D2 [D3]	K		1E	K		1E	D2	210
	D3	L		46	L		C6	L		(46)	L		AC	L		(AC)	L		C6	L		[99]	L		09	L		33	L		09	L		09	D3	211
	D4	M		49	M		C9	M		(49)	M		AD	M		(AD)	M		C9	M		[8A]	M		07	M		B2 [B3]	M		07	M		07	D4	212
	D5	N		4A	N		CA	N		(4A)	N		AE	N		(AE)	N		CA	N		[92]	N		06	N		72 [73]	N		06	N		06	D5	213
	D6	O		4C	O		CC	O		(4C)	O		AF	O		(AF)	O		CC	O		[9A]	O		03	O		F3	O		03	O		03	D6	214
	D7	P		4F	P		CF	P		(4F)	P		B0	P		(B0)	P		CF	P		[8B]	P		0D	P		0A [0B]	P		0D	P		0D	D7	215

Ref.	EBCDIC			IBM 1030			IBM 1050			IBM 1060			IBM 2260 – 2260			IBM 2260 – 1053			IBM 2740			IBM 7770/7772			AT&T 83B3 W U 115A			WU TWX				Charac
	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex) Even	Code (Hex) Non	Graphic
	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
216	Q		D8	Q		51	Q		D1	Q		(51)	Q		B1	Q		(B1)	Q		D1	Q		[93]	Q		1D	Q		8B	8B	Q
217	R		D9	R		52	R		D2	R		(52)	R		B2	R		(B2)	R		D2	R		[9B]	R		0A	R		4B	4B	R
218			DA																													
219			DB																													
220			DC																													
221			DD																													
222			DE																													
223			DF																													
224		RM	E0					RM	34																							
225			E1[3]																									\[3]		3A	3B	
226	S		E2	S		25	S		A5	S		(25)	S		B3	S		(B3)	S		A5	S		[51]	S		14	S		CA	CB	S
227	T		E3	T		26	T		A6	T		(26)	T		B4	T		(B4)	T		A6	T		[59]	T		01	T		2B	2B	T
228	U		E4	U		29	U		A9	U		(29)	U		B5	U		(B5)	U		A9	U		[4A]	U		1C	U		AA	AB	U
229	V		E5	V		2A	V		AA	V		(2A)	V		B6	V		(B6)	V		AA	V		[52]	V		0F	V		6A	6B	V
230	W		E6	W		2C	W		AC	W		(2C)	W		B7	W		(B7)	W		AC	W		[5A]	W		19	W		EB	EB	W
231	X		E7	X		2F	X		AF	X		(2F)	X		B8	X		(B8)	X		AF	X		[4B]	X		17	X		1B	1B	X
232	Y		E8	Y		31	Y		B1	Y		(31)	Y		B9	Y		(B9)	Y		B1	Y		[53]	Y		15	Y		9A	9B	Y
233	Z		E9	Z		32	Z		B2	Z		(32)	Z		BA	Z		(BA)	Z		B2	Z		[5B]	Z		11	Z		5A	5B	Z
234			EA																													
235			EB																													
236			EC																													
237			ED																													
238			EE																													
239			EF																													
240	0		F0	0[4]		(15) [20]	0		15	0		15	0		50	0		(50)	0		15	0		[14] [40]	0		2D	0		DC	0D	0
241	1		F1	1		02	1		02	1		02	1		51	1		(51)	1		02	1		[09]	1		3D	1		8D	8D	1
242	2		F2	2		04	2		04	2		04	2		52	2		(52)	2		04	2		[11]	2		39	2		4D	4D	2
243	3		F3	3		07	3		07	3		07	3		53	3		(53)	3		07	3		[19]	3		30	3		CC	CD	3
244	4		F4	4		08	4		08	4		08	4		54	4		(54)	4		08	4		[0A]	4		2A	4		2D	2D	4
245	5		F5	5		0B	5		0B	5		0B	5		55	5		(55)	5		0B	5		[12]	5		21	5		AC	AD	5
246	6		F6	6		0D	6		0D	6		0D	6		56	6		(56)	6		0D	6		[1A]	6		35	6		6C	6D	6
247	7		F7	7		0E	7		0E	7		0E	7		57	7		(57)	7		0E	7		[0B]	7		3C	7		ED	ED	7
248	8		F8	8		10	8		10	8		10	8		58	8		(58)	8		10	8		[13]	8		2C	8		1D	1D	8
249	9		F9	9		13	9		13	9		13	9		59	9		(59)	9		13	9		[1B]	9		23	9		9C	9D	9
250			FA																													
251			FB																													
252			FC																													
253			FD																													
254			FE																													
255			FF																													

C		IBM 1030			IBM 1050			IBM 1060			IBM 2260: 2260			IBM 2260: 1053			IBM 2740			IBM 7770/7772			AT&T 83B3 W U 115A			WU TWX			WTTA (ITA2)			WTTA (ZSC3)			EBCDIC	
trol	Code (Hax)	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex) Even Non	Character Graphic	Character Control	Code (Hex)	Character Graphic	Character Control	Code (Hex)	Code (Hex)	Ref.
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	
	D8	Q		51	Q		D1	Q		(51)	Q		B1	Q		(B1)	Q		D1	Q		[93]	Q		1D	Q		8B 8B	Q		1D	Q		1D	D8	216
	D9	R		52	R		D2	R		(52)	R		B2	R		(B2)	R		D2	R		[9B]	R		0A	R		4B 4B	R		0A	R		0A	D9	217
	DA																																		DA	218
	DB																																		DB	219
	DC																																		DC	220
	DD																																		DD	221
	DE																																		DE	222
	DF																																		DF	223
M	E0					RM	34																												E0	224
	E1[3]																									\[3]		3A 3B							E1	225
	E2	S		25	S		A5	S		(25)	S		B3	S		(B3)	S		A5	S		[51]	S		14	S		CA CB	S		14	S		14	E2	226
	E3	T		26	T		A6	T		(26)	T		B4	T		(B4)	T		A6	T		[59]	T		01	T		2B 2B	T		01	T		01	E3	227
	E4	U		29	U		A9	U		(29)	U		B5	U		(B5)	U		A9	U		[4A]	U		1C	U		AA AB	U		1C	U		1C	E4	228
	E5	V		2A	V		AA	V		(2A)	V		B6	V		(B6)	V		AA	V		[52]	V		0F	V		6A 6B	V		0F	V		0F	E5	229
	E6	W		2C	W		AC	W		(2C)	W		B7	W		(B7)	W		AC	W		[5A]	W		19	W		EB EB	W		19	W		19	E6	230
	E7	X		2F	X		AF	X		(2F)	X		B8	X		(B8)	X		AF	X		[4B]	X		17	X		1B 1B	X		17	X		17	E7	231
	E8	Y		31	Y		B1	Y		(31)	Y		B9	Y		(B9)	Y		B1	Y		[53]	Y		15	Y		9A 9B	Y		15	Y		15	E8	232
	E9	Z		32	Z		B2	Z		(32)	Z		BA	Z		(BA)	Z		B2	Z		[5B]	Z		11	Z		5A 5B	Z		11	Z		11	E9	233
	EA																																		EA	234
	EB																																		EB	235
	EC																																		EC	236
	ED																																		ED	237
	EE																																		EE	238
	EF																																		EF	239
	F0	0[4]		(15) [20]	0		15	0		15	0		50	0		(50)	0		15	0		[14] [40]	0		2D	0		DC 0D	0		2D	0		2B	F0	240
	F1	1		02	1		02	1		02	1		51	1		(51)	1		02	1		[09]	1		3D	1		8D 8D	1		3D	1		3C	F1	241
	F2	2		04	2		04	2		04	2		52	2		(52)	2		04	2		[11]	2		39	2		4D 4D	2		39	2		3A	F2	242
	F3	3		07	3		07	3		07	3		53	3		(53)	3		07	3		[19]	3		30	3		CC CD	3		30	3		39	F3	243
	F4	4		08	4		08	4		08	4		54	4		(54)	4		08	4		[0A]	4		2A	4		2D 2D	4		2A	4		36	F4	244
	F5	5		0B	5		0B	5		0B	5		55	5		(55)	5		0B	5		[12]	5		21	5		AC AD	5		21	5		35	F5	245
	F6	6		0D	6		0D	6		0D	6		56	6		(56)	6		0D	6		[1A]	6		35	6		6C 6D	6		35	6		33	F6	246
	F7	7		0E	7		0E	7		0E	7		57	7		(57)	7		0E	7		[0B]	7		3C	7		ED ED	7		3C	7		27	F7	247
	F8	8		10	8		10	8		10	8		58	8		(58)	8		10	8		[13]	8		2C	8		1D 1D	8		2C	8		2E	F8	248
	F9	9		13	9		13	9		13	9		59	9		(59)	9		13	9		[1B]	9		23	9		9C 9D	9		23	9		2D	F9	249
	FA																																		FA	250
	FB																																		FB	251
	FC																																		FC	252
	FD																																		FD	253
	FE																																		FE	254
	FF																																		FF	255

APPENDIX I: TRANSMISSION CODE CHART

This chart may be used in reading transmission code as found in main storage. In the leftmost column of each section of the chart is shown the hexadecimal representations of the 256 bit patterns that can appear in a System/360 byte. Opposite each bit pattern is the character represented by that bit pattern in the EBCDIC character set and in the character sets for each of the types of terminals listed in the remaining columns of the chart. (The specific character set options represented in the chart are the same as those listed in Appendix H.) For example, before translation to EBCDIC a hexadecimal 04 appearing in main storage would represent the digit 2, if the bit pattern was received from an IBM 1030, 1050, 1060, or 2740; the character EOT, if from a 2260; or a Space character, if from an 83B3, 115A, or World Trade telegraph terminal. The absence of a character in the column headed AT&T TWX signifies that the bit pattern 04 is undefined for TWX terminals.

Example: In order to translate

1601E4CC A5011515 150201CA B1E70190

as found in storage, first separate the characters into pairs:

16 01 E4 CC A5 01 15 15

15 02 01 CA B1 E7 01 90

If this sequence was received from an IBM 1050, it represents the characters:

EOA SP B O S SP 0 0

0 1 SP N Y C SP *

so that the message entered at the 1050 terminal was, in part,

BOS 0001 NYC *

S/360 Byte (Hex)	EBCDIC		IBM 1030		IBM 1050		IBM 1060		IBM 2260: 2260		IBM 2260: 1053		IBM 2740		IBM 7770, 7772		AT&T 83B3 WU 115A		WU TWX		WTTA: ITA2		WTTA: ZSC3		S/360 Byte (Hex)
	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	
00		NUL																							00
01		SOH		SP		SP		SP				SOH		SP			T				T		T		01
02		STX	1		1		1			STX		STX	1					CR				CR		CR	02
03		ETX								ETX		ETX					O		@		O		O		03
04		PF	2		2		2			EOT		EOT	2					SP				SP		SP	04
05		HT															H			SP	H		H		05
06		LC								ACK		ACK					N				N		N		06
07		DEL	3		3		3						3				M				M		M		07
08			4		4		4						4					LF				LF		LF	08
09															1		L				L		L		09
0A		SMM							◢	NL		NL			4		R		P		R		R		0A
0B		VT	5		5		5						5		7		G		P		G		G		0B
0C		FF														SP	I				I		I		0C
0D		CR	6		6		6						6				P		0		P		P		0D
0E		SO	7		7		7						7				C				C		C		0E
0F		SI															V				V		V		0F
10		DLE	8		8		8						8				E				E		E		10
11		DC1													2		Z				Z		Z		11
12		DC2													5		D		H		D		D		12
13		TM	9		9		9						9		8		B		H		B		B		13
14		RES													0		S		(		S		S		14
15		NL	0		0		0			NAK		NAK	0				Y		(		Y		Y		15
16		BS	#	EOA	#	EOA	#	EOA					#	EOA			F				F		F		16
17		IL															X				X		X		17
18		CAN								CAN							A				A		A		18
19		EM				PN									3		W				W		W		19
1A		CC				RS									6		J				J		J		1A
1B		CU1													9			FIGS	X			FIGS		FIGS	1B
1C		IFS				Upshift								Upshift		CU3	U				U		U		1C
1D		IGS															Q		8		Q		Q		1D
1E		IRS															K				K		K		1E
1F		IUS		EOT		EOT		EOT						EOT				LTRS				LTRS		LTRS	1F
20		DS	s@ H:		@			Add					@												20
21		SOS														CU2	5			EOT	5		•		21
22		FS														CU1			D			CR		CR	22
23			/		/		/						/			EOT	9		D		9		:		23
24		BYP														EOB		SP	$			SP		SP	24
25		LF	S		s		S						s				#		$				?		25
26		ETB (EOB)	T		t		T						t				A' C7/8				,		,		26
27		ESC(PRE)															•				•		7		27
28																				TpAuxOff		LF		LF	28
29			U		u		U						u				A) C3/4			TpAuxOff	)		)		29
2A		SM	V		v		V						v				4				4		/		2A
2B		CU2															& •		T				0		2B
2C			W		w		W						w				8				8			Bell	2C
2D		ENQ															0		4		0		9		2D
2E		ACK															A: C1/8				:		8		2E
2F		BEL	X		x		X						x				A; C3/8				=		=		2F
30																	3			FF	3		-		30
31			Y		y		Y						y				"			FF	+				31
32		SYN	Z		z		Z						z				$					WRU		WRU	32
33																	A? C5/8		L		?		6		33
34		PN				RM											A' CBell				'		'		34
35		RS															6		,		6		5		35
36		UC															C1/4						4		36
37		EOT	,	(S)	,		,						,	(S)			/				/				37
38						BYP											-				-		+		38
39																	2				2		3		39
3A																	ABell		\		c"	Bell	2		3A
3B		CU3		LF		LF		LF						LF				FIGS				FIGS		FIGS	3B
3C		DC4															7		<		7		1		3C
3D		NAK		EOB		EOB		EOB						EOB			1		<		1				3D
3E						PRE											A(C1/2				(		(		3E
3F		SUB																LTRS				LTRS		LTRS	3F

S/360 Byte (Hex)	EBCDIC		IBM 1030		IBM 1050		IBM 1060		IBM 2260 R: 2260		IBM 2260 R: 1053		IBM 2740		IBM 7770, 7772		AT&T 83B3 WU 115A		WU TWX		WTTA: ITA2		WTTA: ZSC3		S/360 Byte (Hex)
	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	
40		SP	-	(N)	-	(N)	-	(N)		SP		SP	-	(N)	0										40
41									—	EOM	!														41
42									|	Check	"								B						42
43			J		j		J		#		#		j						B						43
44									$		$								"						44
45			K		k		K		%		%		k						"						45
46			L		l		L		&		&		l												46
47									'		'														47
48									(		(									TpAuxOn					48
49			M		m		M		)		)		m		/					TpAuxOn					49
4A	¢		N		n		N		*		*		n		U										4A
4B	.								+		+				X				R						4B
4C	<		O		o		O		,		,		o												4C
4D	(								-		-								2						4D
4E	+								.		.														4E
4F	|		P		p		P		/		/		p												4F
50	&								0		0									LF					50
51			Q		q		Q		1		1		q		S					LF					51
52			R		r		R		2		2		r		V										52
53									3		3				Y				J						53
54						MZ		Message	4		4														54
55									5		5								*						55
56									6		6														56
57			$		$		$		7		7		$												57
58						RES	★		8		8														58
59									9		9				T										59
5A	!								:		:				W				Z						5A
5B	$			LF-CR		NL		CR	;		;			NL	Z				Z						5B
5C	*								<		<								:						5C
5D	)					BS			=		=			BS					:						5D
5E	;					IL		IL	>		>			IL											5E
5F	¬								?		?														5F
60	-																								60
61	/		s& H+		&		+						&												61
62			A		a		A						a												62
63																			F						63
64			B		b		B						b												64
65																			&						65
66																									66
67			C		c		C						c												67
68			D		d		D						d												68
69																									69
6A		EOM																	V						6A
6B	,		E		e		E						e						V						6B
6C	%																		6						6C
6D	_		F		f		F						f						6						6D
6E	>		G		g		G						g												6E
6F	?																								6F
70			H		h		H						h												70
71																				SO					71
72																			N						72
73			I		i		I						i						N						73
74																			•						74
75						PZ		Restore											•						75
76			.	(Y) EOFC	.	(Y)	.	(Y)					.	(Y)											76
77																									77
78																									78
79						PF		Subtr																	79
7A	:			HT		Tab		Tab						HT											7A
7B	#	EOA																	↑						7B
7C	@					Dwnshft								Dwnshft											7C
7D	'																		>						7D
7E	=																								7E
7F	"			EOC DEL		DEL		DEL						DEL											7F

S/360 Byte (Hex)	EBCDIC		IBM 1030		IBM 1050		IBM 1060		IBM 2260				IBM 2740		IBM 7770, 7772		AT&T 83B3 W U 115A		WU TWX		WTTA				S/360 Byte (Hex)
									2260		1053										ITA2		ZSC3		
	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Ctl	Gr	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	
80															-(7770 only)										80
81	a					SP								SP											81
82	b				=								=						A						82
83	c																		A						83
84	d				<								<						!						84
85	e																		!						85
86	f																								86
87	g				;								;												87
88	h				:								:							X-On					88
89	i														J					X-On					89
8A															M										8A
8B					%								%		P				Q						8B
8C																									8C
8D					'								'						1						8D
8E					>								>												8E
8F																									8F
90					*								*							HT					90
91	j														K					HT					91
92	k														N										92
93	l				(								(		Q				I						93
94	m																								94
95	n				)								)						)						95
96	o				"	EOA							"	EOA											96
97	p																								97
98	q																								98
99	r					PN									L										99
9A						RS									O				Y						9A
9B															R				Y						9B
9C						Upshift								Upshift					9						9C
9D																			9						9D
9E																									9E
9F																									9F
A0					¢								¢							WRU					A0
A1									A		A									WRU					A1
A2	s								B		B														A2
A3	t				?				C		C		?						E						A3
A4	u								D		D														A4
A5	v				S				E		E		S						%						A5
A6	w				T				F		F		T												A6
A7	x								G		G														A7
A8	y								H		H														A8
A9	z				U				I		I		U												A9
AA					V				J		J		V						U						AA
AB									K		K								U						AB
AC					W				L		L		W						5						AC
AD									M		M								5						AD
AE									N		N														AE
AF					X				O		O		X												AF
B0									P		P														B0
B1					Y				Q		Q		Y							CR					B1
B2					Z				R		R		Z						M						B2
B3									S		S								M						B3
B4									T		T								-						B4
B5									U		U								-						B5
B6									V		V														B6
B7					|				W		W		|											B7
B8						BYP			X		X														B8
B9									Y		Y														B9
BA									Z		Z														BA
BB						LF								LF					]						BB
BC																									BC
BD						EOB			▶Start MI		¢			EOB					=						BD
BE						PRE																			BE
BF									–		–														BF

S/360 Byte (Hex)	EBCDIC		IBM 1030		IBM 1050		IBM 1060		IBM 2260 R				IBM 2740		IBM 7770, 7772		AT&T 83B3 W U 115A		WU TWX		WTTA				S/360 Byte (Hex)
									2260		1053										ITA2		ZSC3		
	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	Gr	Ctl	
C0		PZ			_	N							_	N	& (7770 only)										C0
C1	A																								C1
C2	B																								C2
C3	C				J								J						C						C3
C4	D																								C4
C5	E				K								K						#						C5
C6	F				L								L												C6
C7	G																								C7
C8	H																								C8
C9	I				M								M		A					X-Off					C9
CA					N								N		D				S						CA
CB															G				S						CB
CC					O								O						3						CC
CD																			3						CD
CE																									CE
CF					P								P												CF
D0		MZ																							D0
D1	J				Q								Q		B					VT					D1
D2	K				R								R		E				K						D2
D3	L														H				K						D3
D4	M																		+						D4
D5	N																		+						D5
D6	O																								D6
D7	P				|								|												D7
D8	Q					RES																			D8
D9	R														C										D9
DA															F										DA
DB						NL								NL	I				[						DB
DC																			0						DC
DD						BS								BS					;						DD
DE						IL								IL											DE
DF				Pad																					DF
E0		RM							@		@														E0
E1					+								+							Bell					E1
E2	S				A								A						G						E2
E3	T																		G						E3
E4	U				B								B						'						E4
E5	V																		'						E5
E6	W																								E6
E7	X				C								C												E7
E8	Y				D								D												E8
E9	Z																								E9
EA																									EA
EB					E								E						W						EB
EC																									EC
ED					F								F						7						ED
EE					G								G												EE
EF																									EF
F0	0				H								H												F0
F1	1																			SI					F1
F2	2																								F2
F3	3				I								I						O						F3
F4	4																								F4
F5	5																		/						F5
F6	6				¬	Y							¬	Y											F6
F7	7																								F7
F8	8																								F8
F9	9					PF																			F9
FA						Tab								HT					←						FA
FB																			←						FB
FC						Dwnshft			¬		¬			Dwnshft					?						FC
FD																			?	SI					FD
FE									|		|													FE
FF						DEL								DEL						Rubout					FF

Start-Stop Communications

The types of remote start-stop (asynchronous) terminals that can communicate with a System/360 under BTAM control, and the kinds of communication lines that can be controlled, are described below. The communication lines must be connected to the computer via an IBM 2701 Data Adapter Unit, and IBM 2702 Transmission Control, or an IBM 2703 Transmission Control.

1. Nonswitched lines (point-to-point or multipoint), using programmed polling:

   IBM 1030 Data Collection System
   IBM 1050 Data Communications System
   IBM 1060 Data Communications System
   IBM 2260 Display Station -- IBM 2848 Display Control (Remote -- 2701 only)
   IBM 2265 Display Station -- IBM 2845 Display Control (Remote -- 2701 only)
   IBM 2740 Communications Terminal (Model 1): Basic: with checking[1]; with Station Control[2]; with Checking and Station Control[2]; or with Checking and IBM 2760 Optical Image Unit features (point-to-point only, if 2740 is equipped with 2760 Optical Image Unit)
   (Model 2): Basic or with Checking[1]
   IBM 2741 Communications Terminal
   Western Union Plan 115A Outstations
   AT&T 83B3 Selective Calling Stations

2. Switched lines:
   IBM 1050 Data Communications System
   IBM 2740 Communications Terminal (Model 1): Dial; Dial, with Checking; Dial, with Transmit Control; Dial, with Checking and Transmit Control, or Dial, with Checking and IBM 2760 Optical Image Unit features.
   IBM 2741 Communications Terminal
   WU Model 33/35 Teletypewriter Exchange Terminal (TWX)

3. Nonswitched multipoint lines using the Auto Poll facility (IBM 2702 or 2703 only):

   IBM 1030 Data Collection System
   IBM 1050 Data Communications System
   IBM 1060 Data Communications System
   IBM 2740 (Model 1): with Station Control[2] or with Station Control[2] and Checking features

----
[1]Used as a regular terminal or as an operator's console, when the operating system includes the Multiple Console Support.

Binary Synchronous Communications

The types of remote binary synchronous stations (computers or terminals) that can communicate with a central System/360 under BTAM control, and the kinds of communications lines that can be controlled, are as follows. The communications lines must be connected to the central computer via an IBM 2701 Data Adapter Unit or an IBM 2703 Transmission Control. An IBM 2701 (with Synchronous Data Adapter Type II) may be attached to either the multiplexer channel or a selector channel. An IBM 2703 (with Synchronous Base Type 1) must be attached to the multiplexer channel.

1. Nonswitched point-to-point and switched point-to-point lines:

   IBM System/360[3]

   IBM System/360 Model 20

   IBM System/3

   IBM 1130 Computing System

   IBM 1800 Data Acquisition and Control System

   IBM 2715 Transmission Control Unit (Model 1 attaches directly to multiplexer channel of central computer; Model 2 communicates with central computer via IBM 2701 or 2703)

   IBM 2770 Data Communications System

   IBM 2780 Data Transmission Terminal

   IBM 3735 Programmable Buffered Terminal

2. Nonswitched multipoint lines:

----
[2]Station Control feature cannot be used if the 2740 is also used as a console under Multiple Control Support.
[3]The remote System/360 may be a Model 25, 30, 40, 50, 65, 67 (operating in 65 mode), 75, 85, or 91.

IBM System/360 Model 20

IBM System/3

IBM 1130 Computing System

IBM 1800 Data Acquisition and Control System

IBM 2715 Transmission Control Unit (Model 1 attaches directly to multiplexer channel of central computer; Model 2 communicates with central computer via IBM 2701 or 2703)

IBM 2770 Data Communications System

IBM 2780 Data Transmission Terminal

IBM 2972 (Models 8 & 11) General Banking Terminal System

Remote IBM 3270 Display System

IBM 3735 Programmable Buffered Terminal (Requires special feature)

LOCAL COMMUNICATIONS

The local IBM 3270 Display System can communicate with a System/360 under BTAM control. The local 3270 display system is connected to the computer by means of a selector, multiplexer, or block multiplexer channel.

Machine and Programming Requirements

BTAM operates on any System/360 that meets the following requirements:

- The system must meet the minimum configuration of the System/360 Operating System.
- The remote stations must be attached to an acceptable data adapter or transmission control unit (IBM 2701, 2702, or 2703). (A local 2715 (Model 1) must be connected to the multiplexer channel.)
- All remote start-stop terminals that are connected to the same multipoint line, or are capable of communicating with the computer over any given switched line termination must be of the same type and must be equipped with the same features. (Remote binary synchronous stations are not subject to this limitation.)
- All devices must be attached to the System/360 via the multiplexer channel except the IBM 2701 with Synchronous Data Adapter Type II, which may be attached via the selector channel (non-switched lines only), or to the multiplexer channel.
- No device may be operated in burst mode concurrently with the operation of BTAM except the 2701 attached via the selector channel.
- Execution of BTAM requires that the interval timer of the central computer be working.
- In a system in which BTAM is used in more than one partition, if the BTAM Read/Write module (IGG019MA) is resident, all device I/O modules that are shared by the BTAM-using partitions must also be made resident.
- Use of the STIMER macro by the user is restricted during the time a BTAM Open (OPEN) or Line Open (LOPEN) operation is in progress, because the BTAM Open routines also use STIMER.

> Terminals which are equivalent to those explicitly supported may also function satisfactorily. The customer is responsible for establishing equivalency. IBM assumes no responsibility for the impact that any changes to the IBM-supplied products or programs may have on such terminals.

These charts show for each transmission code bit pattern the corresponding 2980 character, for each of the models of the 2980 (1, 2, and 4). Also shown is the EBCDIC character equivalent for that bit pattern.

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
00	NUL						
01	SOH						
02	STX	STX	STX	STX	STX	STX	STX
03	ETX	ETX	ETX	ETX	ETX	ETX	ETX
04	PF			open chute			open chute
05	HT	HT	HT	HT	HT	HT	HT
06	LC				LC	LC	LC
07	DEL						
08							
09	RLF						
0A	SMM						
0B	VT						
0C	FF						
0D	CR						
0E	SO						
0F	SI						
10	DLE	DLE	DLE	DLE	DLE	DLE	DLE
11	DC1						
12	DC2						
13	TM						
14	RES			turn page light			turn page light
15	NL	NL	NL	NL	NL	NL	NL
16	BS						
17	IL	message light		message light	message light		message light
18	CAN						
19	EM						
1A	CC						
1B	CU1						
1C	IFS						
1D	IGS						
1E	IRS						
1F	IUS						
20	DS						
21	SOS						
22	FS						
23							
24	BYP*	BYP	BYP	BYP			
25	LF	pass-book index		pass-book index	pass-book index		pass-book index
26	ETB	ETB	ETB	ETB	ETB	ETB	ETB
27	ESC						
28							
29							
2A	SM						
2B	CU2						
2C							
2D	ENQ	ENQ	ENQ	ENQ	ENQ	ENQ	ENQ
2E	ACK						
2F	BEL						
30							
31							
32	SYN	SYN	SYN	SYN	SYN	SYN	SYN
33							
34	PN*						
35	RS						
36	UC	UC	UC	UC			
37	EOT	EOT	EOT	EOT	EOT	EOT	EOT
38							

* Also used as a Terminal Selection Character.

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
39							
3A							
3B	CU3						
3C	DC4						
3D	NAK	NAK	NAK	NAK	NAK	NAK	NAK
3E							
3F	SUB						
40	SP*	SP	SP	SP	SP	SP	SP
41							
42							
43							
44							
45							
46							
47							
48							
49							
4A	¢						
4B	.	3	.		.	¬	
4C	<						
4D	(						
4E	+						
4F	|						
50	&	validate I.D. char.	&	validate I.D. char	&	+	&
51							
52							
53							
54							

* Also used as a Terminal Selection Character

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
55							
56							
57							
58							
59							
5A	!						
5B	$	-	$		$	!	
5C	*	$	&		*	¢	
5D	)						
5E	;						
5F	¬						
60	-	F	-		-	_	
61	/	T	/		/	?	
62							
63							
64							
65							
66							
67							
68							
69							
6A							
6B	,	2	,	,	,	|	
6C	%						
6D	_						
6E	>						
6F	?						
70							
71							

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
72							
73							
74							
75							
76							
77							
78							
79							
7A	:						
7B	#	$	#	‡	#	"	⧻
7C	&						
7D	'						
7E	=						
7F	"						
80							
81	a						
82	b						
83	c						
84	d						
85	e						
86	f						
87	g						
88	h						
89	i						
8A							
8B							
8C							
8D							
8E							

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
8F							
90							
91	j						
92	k						
93	l						
94	m						
95	n						
96	o						
97	p						
98	q						
99	r						
9A							
9B							
9C							
9D							
9E							
9F							
A0							
A1							
A2	s						
A3	t						
A4	u						
A5	v						
A6	w						
A7	x						
A8	y						
A9	z						
AA							
AB							

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
AC							
AD							
AD							
AF							
B0							
B1							
B2							
B3							
B4							
B5							
B6							
B7							
B8							
B9							
BA							
BB							
BC							
BD							
BE							
BF							
C0							
C1	A	C	a	A	A	A	A
C2	B		b	B	B	B	B
C3	C	ł	c	C	C	C	C
C4	D	$\overset{N}{B}$	d	?	D	D	D
C5	E	X	e	E	E	E	E
C6	f	$\overset{O}{B}$	f	F	F	F	F
C7	G	$\overline{S}$	g	G	G	G	G

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
C8	H	$\underline{00}$	h	#	H	H	6
C9	I	8	i	$\bar{C}$	I	I	0
CA							
CB*				M			2
CC	⎍						
CD							
CE	⑂						
CF							
D0*							R
D1	J	4	j	J	J	J	J
D2	K	5	k	K	K	K	K
D3	L	6	l	L	L	L	Q
D4	M	1	m	X	M	M	M
D5	N	0	n	N	N	N	N
D6	O	9	o	O	O	O	I
D7	P	+	p	P	P	P	H
D8	Q	R	q	$\overset{O}{B}$	Q	Q	5
D9	R	$\bar{A}$	r	$\overset{C}{\blacktriangledown}$	R	R	_
DA							
DB							
DC							
DD							
DE							
DF							
E0*				$\overset{\$}{\blacktriangledown}$			4
E1				/			Y
E2	S	$\overset{T}{F}$	s	$\$$	S	S	S

* A non-EBCDIC code.

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
E3	T	$\overline{B}$	t	$\blacktriangle$	T	T	T
E4	U	7	u	$\overline{M}$	U	U	U
E5	V	$\genfrac{}{}{0pt}{}{S}{P}$	v	$\blacktriangledown$	V	V	V
E6	W	Q	w	*	W	W	W
E7	X	$\overset{M}{\blacktriangledown}$	x	$\genfrac{}{}{0pt}{}{N}{B}$	X	X	X
E8	Y	$\overline{D}$	y	$\overset{T}{\blacktriangledown}$	Y	Y	3
E9	Z	V	z		z	z	z
EA							
EB*				I			1
EC							
ED							
EE							
EF							
F0	0	U	0	0	0	)	.
F1	1	$\overset{\blacktriangle}{\blacktriangledown}$	1	1	1	=	L
F2	2	H	2	2	2	<	S
F3	3	$\genfrac{}{}{0pt}{}{C}{F}$	3	3	3	;	#
F4	4	$\overline{M}$	4	4	4	:	O
F5	5	$\overline{L}$	5	5	5	%	P
F6	6	$\overline{C}$	6	6	6	'	*
F7	7	$\genfrac{}{}{0pt}{}{M}{O}$	7	7	7	>	7
F8	8	$\overset{M}{\blacktriangledown}$	8	8	8	*	8
F9	9	$\overset{C}{\blacktriangledown}$	9	9	9	(	9

* A non-EBCDIC code.

8-bit pattern (Hex)	EBCDIC character	2980 character					
		Numeric shift			Alpha shift		
		Model 1	Model 2	Model 4	Model 1	Model 2	Model 4
FA	LVM						
FB							
FC							
FD							
FE	EO						
FF							

GENERAL CHARACTERISTICS

Data received from the IBM 50 Magnetic Data Inscriber[1] (MDI) attachment to the IBM 2772 Multi-Purpose Control Unit contains MDI control characters. The TPEDIT macro enables the user to edit this data. The Edit routine, entered from the TPEDIT macro in the user program, edits the data as specified, then returns control to the user program. You have the option of gaining temporary control (via a user-specified exit routine) to process error records. The Edit routine is written in reentrant code. If data is to be received from more than one MDI at a time, you must provide a separate parameter list for each of them.

When the user program issues a READ macro, it receives one block of data, which may contain one or more MDI logical records (or none). The Edit routine extracts one record from this block of data, edits it and gives it to the user program with a return code indicating whether or not the user program input area is empty. If the input area is not empty, the user program must reissue the TPEDIT macro to obtain another record. When control is returned to the user program with an indication that the input area is empty, the input area can be reused. If the input area contains a partial record, the available portion is edited into the work area, and maintained there. The Edit routine gives a return code indicating that the input area is empty but a record is not available. It is your responsibility to obtain the remainder of the record via READ macros. When control is given back to the Edit routine, the characters in the input area (until EOR is encountered) are treated as the remaining portion of the partial record.

----

[1]For full information on the IBM 50, see the publication IBM 50 Magnetic Data Inscriber Component Description, GA27-2725.

TPEDIT MACRO INSTRUCTION

The TPEDIT macro is used to specify the type of editing to be done on the input received from the IBM 50 MDI attachment to the IBM 2772.

MINLN Specifies the minimum acceptable length of an input record. For EDIT=EDITD, SOR and EOR codes are excluded from the length; for EDITR, SOR and EOR are included in the length.

REPLACE Specifies the code to be used as a replacement character whenever the Edit routine detects a 2772 replacement character (i.e., the EBCDIC SUB character, X'3F') in the input. X'19' is chosen as the assumed value because it is an end-of-data (ED) signal for an IBM 50 MDI cartridge and therefore can never appear as a valid data byte. For REPLACE=X'xx' you specify xx as hexadecimal characters of your choosing. These choices may be made from the code chart in Figure 41, with exceptions as noted below.

Name	Operation	Operand
[name]	TPEDIT	MINLN=n[,REPLACE={<u>X'19'</u> | X'xx'}] [EDIT={<u>EDITD</u> | EDITR}][,RECFM={<u>V</u> | U}] [,ERROPT={<u>IGNORE</u> | name}][,VERCHK={<u>NOCHK</u> | VOKCHK}] [,BUFFER={<u>NO</u> | YES}]

Programming Note: BSC control characters should not be used as replacement characters if the data is to be transmitted via BSC facilities after editing.

Hexadecimal characters representing special purpose MDI codes that should not be used as replacement bytes are:

X'00'(LZ)	X'1E'(VOK)	X'74'(P4)
X'11'(DUP)	X'3C'(RM)	X'75'(P5)
X'12'(LZS)	X'71'(P1)	X'76'(P6)
X'18'(CAN)	X'72'(P2)	X'77'(P7)
X'1D'(GS)	X'73'(P3)	X'78'(P8)

EDIT Specifies the type of editing to be done.

EDITD Causes the input to be edited and start-of-record (SOR) and end-of-record (EOR) delimiters to be deleted.

EDITR Causes the input to be edited and the start-of-record and end-of-record delimiters to be retained as part of the output.

The edit consists of the following functions. Records are extracted one at a time from the input area by scanning for the record delimiting codes (SOR and EOR). DUP codes are replaced by the character from the corresponding location of the record that was in the work area when control was last returned to the Edit routine. (This does not apply to the first record of a cartridge.) Left-zero fields are right-adjusted, with leading zeros inserted where necessary. Left-zero start codes, records containing a cancel code, and group separator codes do not appear in the output stream. Line control characters (ETB, ETX, STX, and DLE STX) are always deleted if found in the input area.

RECFM Specifies the format of the output from the Edit routine. If RECFM=V, a segment descriptor word is appended to each record as shown.

Segment Descriptor Word

nn	bb	Data

Logical Record

where nn (2 bytes) is the length of the logical record and bb (2 bytes) is binary zeros reserved for system use.

This four-byte field is included in the record length returned to the user program via a parameter list.

Note: Allow for this four-byte field when determining the size of the work area (see section on Input to the TPEDIT macro).

If RECFM=U, no segment descriptor word is appended to each record.

ERROPT Specifies whether a user error exit routine is provided to handle erroneous records.

IGNORE An error exit routine is not provided. The error conditions are to be disregarded and the record is to be passed normally to the user program.

name Specifies the name of the user error exit routine to be entered when the Edit routine detects logical errors or replacement characters in the record.

VERCHK (valid only if ERROPT=name is coded) Specifies whether the records are to be checked for verify-OK (VOK) codes. If you specify VOKCHK and a record does not contain the verify-OK code, the record is passed to the error exit routine.

When the Edit routine encounters an erroneous record and control passes to this user-supplied routine, register 13 contains the address of a 72-byte register save area aligned on a fullword boundary, and register 1 contains the address of a two-word parameter list aligned on a fullword boundary. The parameter list is defined as follows:

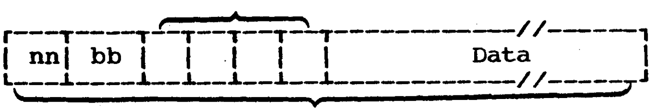
Word Contents

1. Record address
2. Address of record length

The record length includes the four-byte error description word appended, as shown, to the data record. In addition, if RECFM=V, the logical record length (nn) includes these four bytes when the record is passed to the error exit routine.

If RECFM=V is specified:

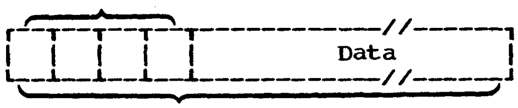
Error
Description Word



Logical record

If RECFM=U is specified:

Error
Description Word



Logical record

Information on the Error Description Word may be found under Error Record Identification. The error exit routine can be used to analyze and, if possible, correct the erroneous record. When control returns to the Edit routine via register 14, you must set register 15 to zero if you wish to bypass the entire record. To direct the Edit routine to pass only that segment of the record in error and process the rest of the record normally, set register 15 to a nonzero value. Note that neither acceptance nor bypassing of the erroneous record changes its effects on subsequent records. The Edit routine removes the error description word when control returns from the error exit routine.

BUFFER Specifies whether or not the user data is in BTAM buffers obtained through dynamic buffering operations. If you specify YES, the Edit routine edits all data in the input area until the area is empty.

Note: The entire buffer chain must have been posted complete in the DECB before you issue the TPEDIT macro.

Input to TPEDIT Macro

Register 1 must point to a four-word parameter list (aligned on a fullword boundary) containing:

Word Contents

1 Input Address
If you are using dynamic buffering, this address points to the first buffer in the chain. The Edit routine edits all records in the buffer chain before indicating that the input area is empty. If dynamic buffering is not used, this is the address of the data to be edited.

2 Input Length
If dynamic buffering is used, this is the length of one buffer. If an I/O area is used, this is the length of the data to be edited.

3 Edit work area address
The work area required by the Edit routine for a given parameter list is obtained in either of two ways. The work area can be provided by the Edit routine (via an unconditional GETMAIN), or you may provide it.

If the work area is to be provided by the EDIT routine, this word must contain binary zeros. The Edit routine issues a GETMAIN macro to obtain the required storage and places the address of the storage obtained in this word. If you provide the work area, this word contains the address of the area supplied. The amount of storage needed in addition to the fixed amount required is determined from:

(1) the maximum record length.
(2) whether a user exit exists (72 bytes for a register save area and 4 bytes for an EDW are required by the macro if an exit is specified).
(3) whether RECFM=V.

The size (in bytes) of the work area may be determined from the formula:

S = 84 + 76E + R + 4V

Where:

S is the size (in bytes) of the work area.

E = 0 if ERROPT=IGNORE is coded

= 1 if ERROPT=NAME is coded

V = 0 if RECFM=U

= 1 if RECFM=V

R is the length of the longest record to be processed.

4 Maximum record length. This is the length, in bytes, of the longest valid edited record. For EDIT=EDITD the length should exclude SOR and EOR codes; for EDIT=EDITR, the length should include SOR and EOR codes.

Bits 0,1		00				01				10				11			
Bits 2,3		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
1st Hex Digit		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	LZ				Space	&	Minus -							$\bar{0}$	0-8-2	0
0001	1		DUP					/	P1					A	J		1
0010	2		LZ Start						P2					B	K	S	2
0011	3								P3					C	L	T	3
0100	4								P4					D	M	U	4
0101	5								P5					E	N	V	5
0110	6								P6					F	O	W	6
0111	7								P7					G	P	X	7
1000	8		CAN						P8					H	Q	Y	8
1001	9		ED											I	R	Z	9
1010	A					¢	!		:								
1011	B						$	,	#								
1100	C				RM	<	*	%	@								
1101	D		GS			(	)	Under-line	' Prime								
1110	E		VOK			+	;	>	=								
1111	F					|	¬	?	"								

Bits 4,5 6,7 — 2nd Hex Digit

IBM 50 Code Name	EBCDIC Code Name
LZ (LZ Fill)	NULL
DUP	DC 1
LZ Start	DC 2
CAN (Cancel)	Can
ED (End Data)	EM
RM (Rec Mark)	DC 4
GS (Group Sep)	IGS
VOK (Ver OK)	IRS

IBM 50 Code Name (See Note)
P1 (Prog 1)
P2 (Prog 2)
P3 (Prog 3)
P4 (Prog 4)
P5 (Prog 5)
P6 (Prog 6)
P7 (Prog 7)
P8 (Prog 8)

Note: Codes are assigned for IBM 50 use only.

Figure 41. IBM 50 MDI Control Codes

The value of the maximum record size should not include the four-byte segment descriptor word added to a variable length record.

Records that exceed the maximum record size are considered erroneous records. Register 13 must contain the address of a 72-byte register save area aligned on a fullword boundary.

Return Codes

After the Edit routine has edited a record, it provides in register 15 a return code indicating record availability and status of the input area, prior to returning control to the user program. The return codes and their meanings are as follows:

Code (hex)	Meaning
00	A record is available; input area is empty. The routine has edited the last logical record in the input area and is passing the record to the user program.
04	A record is available; input area is not empty. The routine has edited one logical record and is passing that record to the user program.
08	No record is available; input area is empty. The last record in the input area was incomplete; i.e., it was a partial record.
0C	End-of-data (ED) code was detected.

For return codes 00 and 04, the record address and the address of the record length are given to the user program in a two-word parameter list aligned on a fullword boundary. The address of the parameter list is returned in register 1. The parameter list has the following format:

Word	Contents
1	Address of the record
2	Address of the record length

IDENTIFICATION OF RECORDS CONTAINING ERRORS

This section describes what the Edit routine considers to be records containing errors. Once the Edit routine has determined a record to be in error, it passes that record to the user error exit routine, if ERROPT=name is specified in the TPEDIT macro statement. If an error exit routine is not specified, the erroneous record is returned to the user program.

The Edit routine maintains information about each record as it is being edited. This information is summarized in the Error Description Word (EDW) described below. When the EDW contains a nonzero value in either the Level Status (byte 0) or the Type Status (byte 1), the record is considered an erroneous record and the EDW is inserted between the four-byte record length field and the data portion if RECFM=V is specified. Otherwise, the EDW is appended to the start of the record to help you analyze the error. Table 49 shows the format of the EDW.

Level Status (Byte 0)

The level status indicator identifies erroneous records that result from interrecord dependency and that cannot be identified in the type status byte. The level status is presented with each erroneous record and has one of the values shown in Table 49.

Table 49. Format of Error Description Word

<u>Byte 0: Level status</u> 0 - for any error record that will not cause questionable data to be in the following records. 1 - for any error record that may cause questionable data to be in the following records. 2 - for any error record that (1) contains questionable data due to the error level of preceding record(s) and (2) may cause questionable data to be in the following records; and where the level status of the previous record was either 1 or 2.
<u>Byte 1: Type status</u> 0 - No identifiable errors. 1 - Start-of-record (SOR) or end-of record (EOR) in error. 2 - Length error. 4 - Field error. 8 - Data check error. Note: This field may contain combinations of these error types; e.g., a C (hexadecimal) indicates a data check error <u>and</u> a field error.
<u>Byte 2: Program Level</u> 1 - P1 5 - P5 E -none of the preceding levels. Start-of-record (SOR) is in error. 2 - P2 6 - P6 3 - P3 7 - P7 4 - P4 8 - P8
<u>Byte 3: Record Status</u> U - Unverified record. V - Verified record. E - Neither U nor V. End-of-record (EOR) is in error.
Note: The error description record is in EBCDIC format. For example, a 2 is represented as X'F2'; a C is represented as X'C3'.

A level status of other than zero is presented with erroneous records resulting from the following:

- The start-of-record (SOR) location has a character defined as an error.
- The record contains two or more data check bytes in succession.
- The record is longer than the user-specified maximum length record.
- The length of the record is not equal to the length of the first valid record of the same program level encountered on the MDI cartridge from which data is being obtained.
- The record has a data duplication dependency on a previous record having one of the foregoing errors.

The level status is set to zero whenever the Edit routine encounters (1) a record without one of the previous errors, (2) a canceled record, or (3) the first record of a cartridge.

Type Status (Byte 1)

The type status indicator identifies records in error because of SOR, EOR, length, field, and/or data check error conditions.

The type status is presented with each erroneous record and has a value of:

0 For any record that has no identifiable errors, but contains questionable data due to a level status of other than zero (see Level Status).

1 For any record that (1) has a SOR character of other than P1 through P8 or a GS code, or (2) has an EOR character of other than a VOK code when you have a specified VERCHK=VOKCHK, or (3) has an EOR character of other than a VOK code or RM code when you have specified VERCHK=NOCHK.

2 For any record that has an incorrect length because it is:

- Longer than the specified maximum, or
- Shorter than the specified minimum (MINLN), or
- Not equal to the length of the first valid record of the same program level encountered on the MDI cartridge from which data is being obtained.

4 For any record that has one or more field errors. A field error is a field or fields where duplication and/or left-zero justification functions did not occur due to an error condition.

8 For any record that has a data check error.

The type status indicator can also have hexadecimal values of 3, 5, 6, 7, 9, A, B, C, D, E and F. These values indicate various combinations of SOR, EOR, length, field, and data check errors. For example, a value of A indicates a record with a data check error (8) as well as an incorrect length error (2).

Note: A data check error is indicated by the presence of 2772 replacement characters (i.e., EBCDIC SUB characters, X'3F'), in the input.

Program Level (Byte 2)

This byte contains an indication of the start-of-record (SOR) character associated with this record. (See Table 49 for values.)

Record Status (Byte 3)

This byte contains an indication of the end-of-record (EOR) character associated with this record. (See Table 49 for values.)

EXAMPLES OF RECORDS CONTAINING ERRORS

Figure 40 shows some of the errors that may occur during processing and their effect on the error description word (EDW). For these records, the maximum record length is specified as 50, EDITR and VOKCHK are specified, and the hexadecimal REPLACE character is '5B' ($). An asterisk in the records indicates the presence of a DUP code in the location before editing.

Record 1 was a valid record. It contained a program level 1 code and thus established the valid length for all program level 1 records received from the cartridge.

Record 2 has a data check in the SOR location. Level status is set to 1 because the SOR location might have contained a cancel code that would cause any data duplicated into the following record to be questionable.

Record 3 has no identifiable error but may contain questionable data because it contained DUP codes and follows a record with a level status of 1.

Record 4 has a data check error. Because it contained no DUP codes, the level status is set to 0.

Record 5 is shorter than first program level 1 records received from the cartridge (length error). This record also contains an RM code rather than a VOK code in the EOR location (VOKCHK was specified). Because the Edit routine cannot determine why the record is short, all data duplicated from this record is questionable; the level status is therefore set to 1.

```
(Record 2)
| 19EV |              ***************        V  |
|      |                                    *O  |
|      | $111378  RECORD NUMBER  2AK            |

(Record 3)
|      |                                     V  |
|      | P        ***************           *O  |
| 201V | 1357987  RECORD NUMBER  3AK            |

(Record 4)
|      |                                     V  |
|      | P                                   O  |
| 081V | 1358977  REC$RD NUMBER  4AK            |

(Record 5)
|      | P                                   R  |
| 131U | 1358436  RECORD NUMBER  5M             |

(Record 6)
|      |                                     V  |
|      | P        ***************           *O  |
| 241V | 1358436  RECORD NUMBER  6$K            |

(Input record 7)
|      |                                                           |     V |
|      | P                                                         |     O |
| 233E | 3998865  RECORD NUMBER  7A  MAXIMUM  00001430  IN  WAREH  | OUSEK |
                       (Error record 7)

(Error record 8)
|      |    V   |
|      |    O   |
| 21EV | OUSEK  |

(Input record 8)
(Error record 9)
|      |                                     V  |
|      | P                                   O  |
| 081V | 1367$82  RECORD NUMBER  8AK            |

Resulting Error
Description Word
```

Figure 42. Examples of Erroneous Records (IBM 50 MDI)

Record 6 contains a DUP code that is beyond the last position of the preceding record.

Record 7 is longer than the maximum specified record length. Note that it is passed as two records. The first record indicates an EOR error and a length error; the second indicates an SOR error.

Record 9 has a data check error. Because it contained no DUP codes, the Level Status is set to zero.

PROGRAMMING CONSIDERATIONS

- All cancelled records are bypassed and are not passed as erroneous records.
- All input records less than three bytes in length (SOR location, one data byte, EOR location) are treated as canceled records. An input record of this size may be the remaining portion of a record that was longer than the maximum user-specified record size.
- Data duplication occurs with the DUP code replaced by the character from the corresponding location of the previous record that was in the work area when control was last returned to the Edit routine.
- For any of the following conditions, data duplication does occur and the DUP code is replaced with the user-specified error replacement character and a field error is indicated:

  The DUP code is encountered in the first record of a cartridge.

  The DUP code is encountered in a record and the previous record was a canceled record.

  The DUP code is encountered in a record and its position would cause duplication of the previous record's end-of-record delineator location or a position beyond the length of the previous record.

  The DUP code is encountered in a record and its position would cause duplication of an error replacement character.
- For either of the following conditions, left-zero justification does not occur, the left-zero-fill code (LZ) is replaced with the user-specified error replace character, and a field error is indicated:

  The left-zero-fill code (LZ) is encountered without its corresponding left-zero-start code (LZS).

  The user-specified maximum record size is exceeded before the valid end of a left-zero field is encountered.
- If dynamic buffering is being used, the BSC control characters ETB and ETX should not be entered as data on IBM 50 MDI cartridges.

END-OF-CARTRIDGE CODE

A unique code, written by the IBM 50 MDI, is used to signal the 2772 control unit that all meaningful data on a cartridge has been read. For the MDI cartridge, the end-of-cartridge code is the ED character (X'19'), which is equivalent to the EBCDIC end-of-medium (EM) character (X'19').

After initiation of a Read operation the MDI continues to read data from the tape until it senses the ED character. When the MDI sends this character to the 2772, the 2772 signals the tape unit to rewind the tape and then transmits the data in its buffer to the central computer.

```
//TEST  JOB  MSGLEVEL=1
//STEP  EXEC  ASMFC
//ASM.SYSLIB DD DSN=SYS1.MACLIB,DISP=OLD
//SYSIN  DD  *
  CONFIGUR CORE=32,PC=YES,INQDISP=YES,GDU=YES,GETID=F0,STORID=F5,        X
               IDCOUNT=8,MONERR=(4,5),FUNCERR=(6,7),ENDERR=(8,9)
*                    THE CORE=32 OPERAND OF THE CONFIGUR MACRO
*                    INDICATES THAT THE 2715 HAS 32K OF STORAGE
*                    AVAILABLE. CODING PC=YES, INDICATES THAT
*                    PULSE COUNTERS EXIST ON THIS 2790 SYSTEM.
*                    INQDISP=YES INDICATES THAT INQUIRY DISPLAY
*                    WILL BE USED ON THE 2790 SYSTEM. GDU=YES
*                    INDICATES THAT 2798 GUIDANCE DISPLAY UNITS
*                    ARE ON THIS 2790 SYSTEM. THE USER CAN
*                    DEFINE EIGHT IDENTIFIERS. THE GET IDENTIFIER
*                    CHARACTER IS THE EBCDIC CHARACTER 0 AND THE
*                    STORE IDENTIFIER CHARACTER IS THE EBCDIC
*                    CHARACTER 5. IF A MONITOR KEY CHECK FAILS,
*                    ERROR GUIDANCE LIGHTS 4 AND 5 WILL BE
*                    TURNED ON AT THE 2798. WHEN AN INVALID
*                    FUNCTION IS RECOGNIZED, ERROR GUIDANCE
*                    LIGHTS 6 and 7 WILL BE TURNED ON AT THE 2798.
*                    WHEN A PREMATURE TERMINATION ERROR OCCURS,
*                    ERROR GUIDANCE LIGHTS 8 AND 9 WILL BE
*                    TURNED ON AT THE 2798.
*
*                    THE FOLLOWING AS MACROS DEFINE 60 AREA
*                    STATIONS WITH ID'S BETWEEN 0 AND 59
*                    FROM WHICH TRANSACTIONS CAN BE ENTERED.
*                    WORKOUT AND NORMAL ARE THE NAMES OF THE
*                    TGROUP MACROS THAT DEFINE THE TRANSACTION
*                    CODES THAT CAN BE USED FROM THE DATA ENTRY
*                    UNITS ON THE SYSTEM. CONTROL IS THE NAME OF
*                    THE TGROUP MACRO THAT DEFINES THE TRANSACTION
*                    CODES THAT CAN BE USED FROM THE AREA STATIONS.
*                    FOR EXAMPLE, THE AS MACRO DEFINING THE AREA
*                    STATION WHOSE ADDRESS IS DECIMALLY REPRESENTED
*                    BY ID=01 INDICATES THAT WORKOUT IS THE NAME
*                    OF THE TGROUP MACRO DEFINING WHICH TRANSACTIONS
*                    CAN BE USED BY THE 32 DATA ENTRY UNITS ON
*                    THIS AREA STATION. THE TRANSACTIONS THAT CAN
*                    BE USED BY THE DATA ENTRY UNITS ARE NOT THE
*                    SAME AS THOSE THAT CAN BE USED BY THE AREA
*                    STATIONS IN THIS TABLE LOAD.
*
*
```

```
AS      ID=00,DEGROUP=(WORKOUT,32)
AS      ID=01,DEGROUP=(WORKOUT,32)
AS      ID=02,DEGROUP=(WORKOUT,4)
AS      ID=03,ASGROUP=CONTROL,DEGROUP=(NORMAL,32)
AS      ID=04,ASGROUP=CONTROL
AS      ID=05,ASGROUP=CONTROL
AS      ID=06,ASGROUP=CONTROL
AS      ID=07,ASGROUP=CONTROL
AS      ID=08,ASGROUP=CONTROL
AS      ID=09,ASGROUP=CONTROL
AS      ID=10,ASGROUP=CONTROL
AS      ID=11,ASGROUP=CONTROL
AS      ID=12,ASGROUP=CONTROL
AS      ID=13,ASGROUP=CONTROL
AS      ID=14,ASGROUP=CONTROL
AS      ID=15,ASGROUP=CONTROL
AS      ID=16,ASGROUP=CONTROL
AS      ID=17,ASGROUP=CONTROL
AS      ID=19,ASGROUP=CONTROL
AS      ID=20,ASGROUP=CONTROL
AS      ID=21,ASGROUP=CONTROL
AS      ID=22,ASGROUP=CONTROL
AS      ID=23,ASGROUP=CONTROL
AS      ID=24,ASGROUP=CONTROL
AS      ID=25,ASGROUP=CONTROL
AS      ID=26,ASGROUP=CONTROL
AS      ID=27,ASGROUP=CONTROL
AS      ID=28,ASGROUP=CONTROL
AS      ID=29,ASGROUP=CONTROL
AS      ID=30,ASGROUP=CONTROL
AS      ID=31,ASGROUP=CONTROL
AS      ID=32,ASGROUP=CONTROL
AS      ID=33,ASGROUP=CONTROL
AS      ID=34,ASGROUP=CONTROL
AS      ID=35,ASGROUP=CONTROL
AS      ID=36,ASGROUP=CONTROL
AS      ID=37,ASGROUP=CONTROL
AS      ID=38,ASGROUP=CONTROL
AS      ID=39,ASGROUP=CONTROL
AS      ID=40,DEGROUP=(WORKOUT1,4)
AS      ID=41,DEGROUP=(WORKOUT1,4)
AS      ID=42,DEGROUP=(WORKOUT1,4)
AS      ID=43,DEGROUP=(WORKOUT1,4)
AS      ID=44,ASGROUP=CONTROL,DEGROUP=(NORMAL,4)
AS      ID=45,ASGROUP=CONTROL,DEGROUP=(NORMAL,4)
AS      ID=46,ASGROUP=CONTROL,DEGROUP=(NORMAL,4)
AS      ID=47,ASGROUP=CONTROL,DEGROUP=(NORMAL,4)
AS      ID=48,DEGROUP=(WORKOUT,4)
AS      ID=49,DEGROUP=(WORKOUT,4)
AS      ID=50,DEGROUP=(WORKOUT,4)
```

```
          AS          ID=51,DEGROUP=(WORKOUT,4)
          AS          ID=52,ASGROUP=CONTROL
          AS          ID=53,ASGROUP=CONTROL
          AS          ID=54,ASGROUP=CONTROL
          AS          ID=55,ASGROUP=CONTROL
          AS          ID=56,DEGROUP=(WORKOUT,4)
          AS          ID=57,DEGROUP=(WORKOUT,4)
          AS          ID=58,DEGROUP=(WORKOUT,4)
          AS          ID=59,ASGROUP=CONTROL
*
*
*                             THE FOLLOWING GDUAS MACROS INDICATE THAT
*                             THIS SYSTEM HAS FOUR AREA STATIONS WITH
*                             2798 GUIDANCE DISPLAY UNITS ATTACHED. THE
*                             GDUNUMB OPERAND INDICATES THE NUMBER OF
*                             2798S ON THAT PARTICULAR AREA STATION.
*                             THERE ARE A TOTAL OF 48 2798 GDUS ON
*                             THIS SYSTEM.
*
*
          GDUAS       ID=00,GDUNUMB=16
          GDUAS       ID=02,GDUNUMB=4
          GDUAS       ID=43,GDUNUMB=16
          GDUAS       ID=44,GDUNUMB=12
*
*                             THE FOLLOWING TGROUP MACROS ASSOCIATE
*                             SPECIFIC TRANSACTION CODES WITH USER
*                             DEFINED TRANSACTIONS. THE TRANSACTION
*                             CODES CAN BE SPECIFIED FROM EITHER DATA
*                             ENTRY UNITS OR FROM AREA STATIONS BUT CAN
*                             NOT BE SPECIFIED FROM 2798 GUIDANCE DISPLAY
*                             UNITS.
*
*
CONTROL  TGROUP TC1=BADGE,TC2=BADGE1,TC3=MANUAL,TC4=CARD,TC5=CARD1,      X
               TC6=CARD2,TC7=BADGE,TC9=(EXPAND,E)
*                             THIS TRANSACTION GROUP IS ENTERED BY THE
*                             2715 WHEN THE USER SELECTS TRANSACTION
*                             CODES 1-7 OR 9 FROM AREA STATIONS WHOSE
*                             ADDRESS IS DECIMALLY REPRESENTED BY IDS
*                             FROM 03-39,44-47,52-55,OR 59. IF
*                             TRANSACTION CODE 1 IS SPECIFIED BY THE
*                             USER AT ONE OF THE ABOVE AREA STATIONS,
*                             THE TRANSACTION DEFINED BY THE TRLIST
*                             MACRO WITH THE NAME BADGE IS ENTERED
*                             BY THE 2715. LIKEWISE, THIS TGROUP MACRO
*                             ASSOCIATES ALL ALLOWABLE TRANSACTION
*                             CODES THAT CAN BE SPECIFIED FROM THE
*                             ABOVE AREA STATION WITH A USER DEFINED
*                             TRANSACTION BEGINNING WITH A TRLIST MACRO.
*                             WHEN TRANSACTION CODE 9 IS SPECIFIED ON
*                             ONE OF THE ABOVE AREA STATIONS, THE USER
*                             MUST SPECIFY ONE MORE TRANSACTION CODES
*                             AS INDICATED BY THE FOLLOWING EXPAND
```

```
*                             TGROUP MACRO. TRANSACTION CODES 91,92,93,
*                             94,95, AND 96 ARE ASSOCIATED WITH A
*                             DIFFERENT TRANSACTION. THIRTEEN
*                             DIFFERENT TRANSACTIONS CAN BE SPECIFIED BY
*                             OPERATORS ON THE ABOVE AREA STATIONS.
*                             CODING E AS IN THE TC9 OPERAND INDICATES
*                             THAT THE ADDITIONAL TRANSACTIONS POINTED
*                             TO BY THE FOLLOWING EXPAND TGROUP MACRO
*                             CAN BE SPECIFIED BY THE OPERATORS ON THE
*                             ABOVE AREA STATIONS.
EXPAND   TGROUP TC1=EXP1,TC2=EXP2,TC3=EXP3,TC4=EXP4,TC5=EXP5,TC6=EXP5
NORMAL   TGROUP TC1=DEU1,TC2=DEU2,TC3=DEU3,TC4=DEU4,TC5=(EXPDEU,E),      X
               TC6-DEU6,TC7=ALARM,TC8=ALRMTX,TC9=TEXT
*                             THIS TRANSACTION GROUP IS ENTERED BY THE
*                             2715 WHEN A TRANSACTION CODE OF 1,2,3,4,51,
*                             52,53,54,6,7,8, OR 9 IS SPECIFIED AT A DATA
*                             ENTRY UNIT ON AN AREA STATION WHOSE ADDRESS
*                             IS DECIMALLY REPRESENTED BY ID=03,44,45,46,
*                             OR 47.
EXPDEU   TGROUP TC1=DEUEXP,TC2=DEUEXP,TC3=DEUEXP,TC4=DEUEXP
WORKOUT  TGROUP  TC1=(ALRMESG,E),TC2=(ALRMESG,E),TC3=EXP,TC4=CPU,        X
               TC5=READ,TC6=READST,TC7=READSID,TC8=DISK,TC9=DISK
WORKOUT1 TGROUP  TC1=RDIPSG,TC2=RDEPSGM,TC3=RDEPSGB,TC4=RDEPGPM,         X
               TC5=RDEPGPB,TC6=RDRSTIP,TC7=RDRSTEPM,                     X
               TC8=(EXPAND1,E),TC9=(EXPAND2,E)
EXPAND1  TGROUP  TC1=RDSTIPM,TC2=RDSTEPM,TC3=RDSTEPB,TC4=RDSIDEPM,       X
               TC5=RDSIDGPM,TC6=RDSIDGPB,TC7=WRIPB,TC8=WREPM,TC9=CPU
ALRMESG  TGROUP  TC1=ALARM,TC2=ALRMTX,TC3=TEXT,TC4=EXPALM,               X
               TC5=EXPALMTX,TC6=EXPTX,TC7=CPU,TC8=CPU,TC9=CPU
EXPAND2  TGROUP  TC1=RDGPEPAA,TC2=RDGPEPBB,TC3=RDSIEPAA,                 X
               TC4=RDMSIEP,TC5=RDSIIPAA,TC6=RDSIIPAB
*
*
*                             THE FOLLOWING ASCTR MACROS DEFINE THE AREA
*                             STATIONS WITH PULSE COUNTERS ATTACHED. ONE
*                             ASCTR MACRO MUST BE CODED FOR EVERY AREA
*                             STATION WITH PULSE COUNTERS ATTACHED.
*
*
    ASCTR    ID=01,HIGHCTR=23,ROUTE=(DISK,LOG)
*                             THIS ASCTR MACRO INDICATES THAT 23 IS THE
*                             HIGHEST COUNTER ON THE AREA STATION WITH
*                             ID=01 ON WHICH EITHER COUNT TESTING OR
*                             SCHEDULE READOUT FUNCTIONS ARE TO BE
*                             PERFORMED BY THE 2715. OVERFLOW AND COUNT
*                             TEST RESPONSE MESSAGES WILL BE ROUTED TO
*                             THE 2715 DISK AND THE 2740.
```

```
*        ASCTR    ID=40,HIGHCTR=00,ROUTE=CPU,NEXTAS=42
*                    THIS ASCTR MACRO INDICATES THAT NO COUNTERS
*                    ON THE AREA STATION WITH ID=40 WILL USE
*                    COUNT TESTING OR SCHEDULE READOUT FUNCTIONS.
*                    OVERFLOW MESSAGES WILL BE TREATED AS
*                    PRIORITY DATA TO BE ROUTED TO THE CPU BY
*                    2715. NEXTAS=42 INDICATES THE NEXT AREA
*                    STATION THAT HAS COUNTERS FOR WHICH COUNT
*                    TESTING OR SCHEDULE READOUT FUNCTIONS WILL
*                    BE PERFORMED HAS ID=42.
         ASCTR    ID=41,HIGHCTR=00,ROUTE=CPU,NEXTAS=42
         ASCTR    ID=42,HIGHCTR=2,ROUTE=(CPU,EXTALRM,ASLOG)
         ASCTR    ID=48,HIGHCTR=00,ROUTE=CPU,NEXTAS=0
*                    CODING NEXTAS=0 IN THIS ASCTR MACRO
*                    INDICATES THAT THERE ARE NO MORE AREA
*                    STATIONS ON THE 2790 SYSTEM THAT HAVE
*                    COUNTERS THAT WILL USE COUNT TESTING OR
*                    SCHEDULE READOUT FUNCTIONS.
         ASCTR    ID=58,HIGHCTR=00,ROUTE=CPU,NEXTAS=0
*
*
*                    THE FOLLOWING CTRGROUP MACROS DEFINE EVERY
*                    COUNTER FOR WHICH COUNT TESTING OR SCHEDULE
*                    READOUT MAY BE PERFORMED.
*
*
         CTRGROUP 1,1,14,ID=01,SROENAB=YES,CTINIT=NCT
*                    THIS CTRGROUP MACRO INDICATES THAT THE
*                    READOUT SCHEDULE USED WILL BE THE FIRST
*                    SCHEDULE (1 MINUTE) DEFINED BY THE
*                    CTRSCHED MACRO FOR COUNTER 1 ON THE AREA
*                    STATION WITH ID=01. THE COUNT TEST
*                    SCHEDULE TO BE USED WILL BE THE FOURTEENTH
*                    SCHEDULE (183 MINUTES) DEFINED BY THE
*                    CTRSCHED MACRO. SROENAB=YES INDICATES THAT
*                    SCHEDULE READOUT WILL BE AUTOMATICALLY
*                    STARTED AT ICPL TIME AT THE 2715 FOR THIS
*                    COUNTER. CTINIT=NCT INDICATES THAT NO COUNT
*                    TESTING WILL BE STARTED AT ICPL TIME BY THE
*                    2715 FOR THIS COUNTER.
         CTRGROUP 2,2,7,ID=01,SROENAB=YES,CTINIT=NCT
         CTRGROUP 3,3,10,ID=01,SROENAB=YES,CTINIT=UNASP
         CTRGROUP 6,13,8,ID=01,CTINIT=UNASP
         CTRGROUP 7,5,9,ID=01,SROENAB=YES
         CTRGROUP 11,9,11,ID=01,SROENAB=YES,CTINIT=NULL
         CTRGROUP 16,4,8,ID=01,CTINIT=NCT
         CTRGROUP 17,6,1,ID=01,SROENAB=YES,CTINIT=NCT
         CTRGROUP 23,7,12,ID=01,SROENAB=YES,CTINIT=NCT
         CTRGROUP 1,1,0,ID=42,SROENAB=YES
         CTRGROUP 2,0,14,ID=42,CTINIT=NCT
```

```
CTRSCHED   1,2,3,4,4,3,2,1,3,1,6,90,83,183
*                    THE CTRSCHED MACRO DEFINES ALL THE SCHEDULES
*                    THAT CAN BE USED FOR SCHEDULE READOUT OR
*                    COUNT TESTING.  EACH SCHEDULE IS DEFINED IN
*                    MINUTES.  FOURTEEN SCHEDULES ARE DEFINED HERE.
*
*                    THE FOLLOWING GDUTRANS MACROS ASSOCIATE USER
*                    DEFINED TRANSACTION CODES FOR THE 2798 GDU
*                    WITH DEFINED TRANSACTIONS. THE TRCODE OPERAND
*                    INDICATES THE TRANSACTION CODE. THE TRLIST
*                    OPERAND INDICATES THE NAME OF THE TRLIST MACRO
*                    THAT DEFINES THE CORRESPONDING TRANSACTION.
*
*
         GDUTRANS TRCODE=00,TRLIST=TEST0
         GDUTRANS TRCODE=01,TRLIST=TEST1
         GDUTRANS TRCODE=02,TRLIST=TEST2
         GDUTRANS TRCODE=03,TRLIST=TEST3
         GDUTRANS TRCODE=04,TRLIST=TEST4
         GDUTRANS TRCODE=05,TRLIST=TEST5
         GDUTRANS TRCODE=06,TRLIST=TEST6
         GDUTRANS TRCODE=07,TRLIST=TEST7
         GDUTRANS TRCODE=08,TRLIST=TEST8
         GDUTRANS TRCODE=10,TRLIST=ROUTE1
         GDUTRANS TRCODE=11,TRLIST=ROUTE2
         GDUTRANS TRCODE=12,TRLIST=TESTJOB1
         GDUTRANS TRCODE=13,TRLIST=TESTJOB2
         GDUTRANS TRCODE=14,TRLIST=TESTJOB3
         GDUTRANS TRCODE=15,TRLIST=CARDORD
         GDUTRANS TRCODE=16,TRLIST=UALMAINT
         GDUTRANS TRCODE=19,TRLIST=INV1
         GDUTRANS TRCODE=20,TRLIST=INV2
         GDUTRANS TRCODE=21,TRLIST=INV3
         GDUTRANS TRCODE=22,TRLIST=INV4
         GDUTRANS TRCODE=23,TRLIST=INV5
         GDUTRANS TRCODE=24,TRLIST=STOCK
         GDUTRANS TRCODE=25,TRLIST=INPROC
         GDUTRANS TRCODE=26,TRLIST=LEADTIME
         GDUTRANS TRCODE=27,TRLIST=RATING
         GDUTRANS TRCODE=28,TRLIST=SUPPLIER
         GDUTRANS TRCODE=29,TRLIST=INTRANS
         GDUTRANS TRCODE=30,TRLIST=LINE
         GDUTRANS TRCODE=31,TRLIST=BIN
         GDUTRANS TRCODE=32,TRLIST=RAWMAT
         GDUTRANS TRCODE=33,TRLIST=ORDER
         GDUTRANS TRCODE=34,TRLIST=QUALCON
         GDUTRANS TRCODE=35,TRLIST=QUOTE
```

```
         GDUTRANS TRCODE=36,TRLIST=LASTPUR
         GDUTRANS TRCODE=37,TRLIST=ECONQTY
         GDUTRANS TRCODE=38,TRLIST=CREDIT
         GDUTRANS TRCODE=39,TRLIST=QUADEQN
*
*
*                  THE FOLLOWING PARAMNUM MACROS ASSOCIATE USER
*                  DEFINED PARAMETER LIST NUMBERS WITH PARAMETER
*                  LISTS TO BE USED BY THE 2715 TO CHECK DISPLAY
*                  ENTRIES FROM THE 2798 GDU.
*                  THE PLN OPERAND DEFINES THE PARAMETER LIST
*                  NUMBER AND THE PARMLST OPERAND INDICATES THE
*                  NAME OF THE PARMLIST MACRO THAT DEFINES THE
*                  CORRESPONDING PARAMETER LIST. THE USER SPECIFIES
*                  WHICH PARAMETER LIST HE WISHES THE 2715 TO USE
*                  WHEN CHECKING A 2798 DISPLAY ENTRY BY CODING
*                  THE PARAMETER LIST NUMBER OF A DEFINED
*                  PARAMNUM MACRO IN THE PARAMNO OPERAND OF THE
*                  GDULIST MACRO.
*
*
         PARAMNUM PLN=01,PARMLST=PAR1
         PARAMNUM PLN=02,PARMLST=PAR2
         PARAMNUM PLN=03,PARMLST=PAR3
         PARAMNUM PLN=04,PARMLST=PAR4
         PARAMNUM PLN=05,PARMLST=PAR5
         PARAMNUM PLN=06,PARMLST=PAR6
         PARAMNUM PLN=07,PARMLST=PAR7
         PARAMNUM PLN=08,PARMLST=PAR8
         PARAMNUM PLN=09,PARMLST=PAR9
         PARAMNUM PLN=10,PARMLST=PAR10
         PARAMNUM PLN=11,PARMLST=PAR11
         PARAMNUM PLN=12,PARMLST=PAR12
         PARAMNUM PLN=13,PARMLST=PAR13
         PARAMNUM PLN=14,PARMLST=PAR14
         PARAMNUM PLN=15,PARMLST=PAR15
         PARAMNUM PLN=16,PARMLST=PAR16
         PARAMNUM PLN=17,PARMLST=PAR17
         PARAMNUM PLN=18,PARMLST=PAR18
         PARAMNUM PLN=19,PARMLST=PAR19
         PARAMNUM PLN=20,PARMLST=PAR20
         PARAMNUM PLN=21,PARMLST=PAR21
         PARAMNUM PLN=22,PARMLST=PAR22
         PARAMNUM PLN=23,PARMLST=PAR23
         PARAMNUM PLN=24,PARMLST=PAR24
         PARAMNUM PLN=25,PARMLST=PAR25
         PARAMNUM PLN=26,PARMLST=PAR26
         PARAMNUM PLN=27,PARMLST=PAR27
```

```
         PARAMNUM PLN=28,PARMLST=PAR28
         PARAMNUM PLN=29,PARMLST=PAR29
         PARAMNUM PLN=30,PARMLST=PAR30
         PARAMNUM PLN=31,PARMLST=PAR31
         PARAMNUM PLN=32,PARMLST=PAR32
         PARAMNUM PLN=33,PARMLST=PAR33
         PARAMNUM PLN=34,PARMLST=PAR34
         PARAMNUM PLN=35,PARMLST=PAR35
         PARAMNUM PLN=36,PARMLST=PAR36
         PARAMNUM PLN=37,PARMLST=PAR37
         PARAMNUM PLN=38,PARMLST=PAR38
         PARAMNUM PLN=39,PARMLST=PAR39
*
*
*                         THE FOLLOWING PARMLIST MACROS GENERATE THE
*                         PARAMETER LISTS TO BE USED BY THE 2715 TO
*                         CHECK DISPLAY ENTRIES FROM THE 2798 GDU.
*
*
PAR1     PARMLIST CKMONKY=NO,TRANSL=YES
*                         THIS MACRO INDICATES THE 2715 WILL NOT
*                         CHECK THE DISPLAY ENTRY TO SEE IF THE MONITOR
*                         KEY IS ON AND THE TRANSLATE FUNCTION MAY BE
*                         USED.
PAR2     PARMLIST CKLNGTH=(17,11)
*                         THIS MACRO INDICATES THAT IF THE LENGTH OF A
*                         2798 GDU ENTRY LENGTH IS NOT 17 CHARACTERS
*                         LONG (16 DATA CHARACTERS PLUS MONITOR KEY BYTE),
*                         THEN OPERATIONAL GUIDANCE LIGHT 11 WILL BE
*                         TURNED ON AT THE 2798 TO INDICATE AN ERROR.
PAR3     PARMLIST CKMONKY=YES,CKMOD11=(15,2,13),IDENT=YES
*                         THIS MACRO INDICATES THAT THE 2715 WILL CHECK
*                         THE MONITOR KEY BYTE IN THE GDU ENTRY TO INSURE
*                         THAT IT WAS ON. IF THE MONITOR KEY IS OFF, THE
*                         ERROR GUIDANCE DEFINED BY THE MONERR OPERAND
*                         OF THE CONFIGUR MACRO WILL BE DISPLAYED ON THE
*                         2798. THE MODULUS 11 CHECK WILL BE
*                         PERFORMED BY THE 2715 IN POSITIONS
*                         15 AND 16. THE CHECK CHARACTER WILL BE IN
*                         POSITION 17. IF THE MODULUS 11 VALUE DOES NOT
*                         EQUAL THE CHECK CHARACTER THEN OPERATIONAL
*                         GUIDANCE LIGHT 13 WILL BE TURNED ON AT THE
*                         2798 TO INDICATE AN ERROR. THE STORE OR GET
*                         IDENTIFIER FUNCTION MAY BE USED IN THIS DATA
*                         ENTRY.
PAR4     PARMLIST CKMONKY=YES,CKMOD10=(15,2,13),IDENT=YES
```

```
PAR5      PARMLIST CKOR=(2,C1,C6,D2,D7,E4),ORGUID=9
*                        THIS MACRO INDICATES THAN AN OR CHECK WILL BE
*                        PERFORMED ON THE CHARACTER IN POSITION 2 OF THE
*                        2798 GDU ENTRY FOR ONE OF THE FOLLOWING EBCDIC
*                        CHARACTERS: A,F,K,P,OR U. IF THE CHARACTER IN
*                        POSITION 2 IS NOT ONE OF THE SPECIFIED
*                        CHARACTERS, THEN OPERATIONAL GUIDANCE LIGHT 9
*                        WILL BE TURNED ON AT THE 2798 TO INDICATE AN
*                        ERROR.
PAR6      PARMLIST CKOR=(3,C2,C7,D3,D8,E5),ORGUID=9
PAR7      PARMLIST CKOR=(4,C3,C8,D4,D9,E6),ORGUID=9
PAR8      PARMLIST CKOR=(5,C4,C9,D5,E2,E7),ORGUID=9
PAR9      PARMLIST CKOR=(6,C5,D1,D6,E3,E8),ORGUID=9
PAR10     PARMLIST CKOR=(7,E9,7B,F1,F6,7C),ORGUID=9
PAR11     PARMLIST CKOR=(8,7F,5A,F2,F7,61),ORGUID=9
PAR12     PARMLIST CKOR=(9,7E,5E,F3,F8,15),ORGUID=9
PAR13     PARMLIST CKOR=(10,7A,5C,F4,F9,25),ORGUID=9
PAR14     PARMLIST CKOR=(11,6F,4E,F5,F0,05),ORGUID=9
PAR15     PARMLIST CKOR=(12,5B,6B),ORGUID=9
PAR16     PARMLIST CKOR=(13,50,4B),ORGUID=9
PAR17     PARMLIST CKOR=(14,60,40),ORGUID=9
PAR18     PARMLIST CKOR=(15,5A),ORGUID=9,                                    *
                CKAND=(2,14,F1,F2,F3,F4,F5,F6,F7,F8,F9,F0,7C,61,15),       *
                ANDGUID=16
*                        THIS MACRO INDICATES THAT AN OR CHECK WILL BE
*                        PERFORMED ON THE CHARACTER IN POSITION 15 OF
*                        THE 2798 ENTRY FOR AN ] CHARACTER. IF IT IS NOT,
*                        OPERATIONAL GUIDANCE LIGHT 9 WILL BE TURNED
*                        ON AT THE 2798 TO INDICATE AN ERROR. ALSO, AN
*                        AND CHECK IS PERFORMED AND THE CHARACTERS IN
*                        POSITIONS 2 THROUGH 14 MUST BE EXACTLY THE
*                        FOLLOWING CHARACTERS: 1,2,3,4,5,6,7,8,9,0,@,1,
*                        NEW LINE. IF THE AND CHECK IS NOT SATISFIED,
*                        OPERATIONAL GUIDANCE LIGHT 16 WILL BE
*                        TURNED ON AT THE 2798 TO INDICATE AN ERROR.
PAR19     PARMLIST CKOR=(16,05),ORGUID=9,                                    *
                CKAND=(2,15,7F,7E,7A,6F,5A,5E,5C,4E,5B,50,60,25,6B,4B),    *
                ANDGUID=16
PAR20     PARMLIST CKOR=(17,4E),ORGUID=9,                                    *
                CKAND=(2,16,D8,D9,E2,E3,E4,E5,E6,E7,E8,E9,7B,40,6B,4B,     *
                05),ANDGUID=16
PAR21     PARMLIST CKAND=(2,17,C1,C2,C3,C4,C5,C6,C7,C8,C9,D1,D2,D3,D4       *
                D5,D6,D7),ANDGUID=16
PAR22     PARMLIST CKNUM=(2,17,12)
*                        THIS MACRO INDICATES THAT CHARACTERS IN
*                        POSITIONS 2 THROUGH 17 MUST BE NUMERIC. IF
*                        ALL THE CHARACTERS IN THE FIELD ARE NOT
*                        NUMERIC, OPERATIONAL GUIDANCE LIGHT 12
*                        WILL BE TURNED ON AT THE 2798 TO INDICATE AN
*                        ERROR.
```

```
PAR23    PARMLIST CKNONUM=(2,17,10)
*                    THIS MACRO INDICATES THAT POSITIONS 2 THROUGH
*                    17 WILL BE CHECKED TO INSURE THAT ALL
*                    CHARACTERS ARE NON-NUMERIC. IF A NUMERIC
*                    CHARACTER IS FOUND IN THE FIELD,
*                    OPERATIONAL GUIDANCE LIGHT 10 WILL BE TURNED ON
*                    AT THE 2798 TO INDICATE AN ERROR.
PAR24    PARMLIST CKRANGE=(2,17,73,00,00,00,00,00,00,00,00,00,00,00,  *
               00,00,00,00),HIGUID=9,LOWGUID=13
PAR25    PARMLIST CKMONKY=YES,CKLNGTH=(8,11),CKNUM=(7,8,12),          *
               CKAND=(2,6,5B,00,00,00,4B),ANDGUID=10
PAR26    PARMLIST CKMONKY=YES,CKLNGTH=(17,11),CKMOD11=(6,11,13),      *
               IDENT=YES,CKMOD10=(5,2,9),CKOR=(8,5C,60,F0,4E,40,      *
               ORGUID=10
PAR27    PARMLIST CKLNGTH=(13,11),CKMOD10=(11,2,13),CKMONKY=YES,      *
               IDENT=YES
PAR28    PARMLIST CKNUM=(2,4,12),CKNONUM=(5,6,10),CKRANGE=(7,8,81,81),*
               HIGUID=9,LOWGUID=13
PAR29    PARMLIST CKLNGTH=(3,11),CKOR=(2,C1,C2,C3,C4,C5),ORGUID=10,   *
               CKNUM=(3,3,11)
PAR30    PARMLIST CKOR=(2,4E,60),ORGUID=10,CKNUM=(3,5,12)
PAR31    PARMLIST CKLNGTH=(6,11),CKRANGE=(2,6,10,09,55,00,00),        *
               HIGUID=9,LOWGUID=13
PAR32    PARMLIST
PAR33    PARMLIST CKLNGTH=(7,11),CKNUM=(2,7,12),IDENT=YES
PAR34    PARMLIST CKLNGTH=(5,11),CKOR=(2,4E,60),ORGUID=10,            *
               CKRANGE=(3,5,10,50,00),                                *
               HIGUID=9,LOWGUID=13,RNGETST=ERROR
PAR35    PARMLIST CKLNGTH=(11,11),CKOR=(4,C4,E3),ORGUID=10,CKAND=(2,3,*
               E2,D3),ANDGUID=10,CKRANGE=(5,11,55,88,00,73,80,39,78), *
               HIGUID=9,LOWGUID=13,RNGETST=ERROR
PAR36    PARMLIST CKLNGTH=(9,11),CKMOD11=(7,2,13),IDENT=YES
PAR37    PARMLIST CKMONKY=YES,IDENT=YES
PAR38    PARMLIST CKRANGE=(2,3,50,90),                                *
               HIGUID=9,LOWGUID=13,RNGETST=ERROR
*                    THIS MACRO INDICATES THAT A RANGE CHECK WILL BE
*                    PERFORMED ON POSITIONS 2 AND 3 TO CHECK THAT
*                    THEIR VALUE LIES BETWEEN 00 AND 59.  IF THE
*                    VALUE OF THE FIELD IS HIGHER THAN 59,
*                    OPERATIONAL GUIDANCE LIGHT 9 IS TURNED ON
*                    AT THE 2798. IF THE VALUE OF THE FIELD IS
*                    LESS THAN 0, OPERATIONAL GUIDANCE LIGHT 13
*                    IS TURNED ON AT THE 2798.  RNGETST=ERROR
*                    INDICATES THAT THE 2715 WILL NOT ACCEPT
*                    RANGE TEST BUT TREATS IT AS AN ERROR.
PAR39    PARMLIST CKLNGTH=(11,11),CKNUM=(8,11,12),
               CKAND=(2,7,F2,F3,F9,F5,F1,F1),ANDGUID=13               *
```

```
*                            THE FOLLOWING DISPGUID MACROS DEFINE THE
*                            DISPLAY GUIDANCE MESSAGES THAT CAN BE
*                            DISPLAYED WHEN A TRANSACTION STEP IS ENTERED.
*                            THE USER INDICATES WHICH MESSAGE HE WANTS
*                            DISPLAYED AT THE 2798 FOR A STEP BY CODING
*                            THE NAME OF A DISPGUID MACRO IN THE DISPMSG
*                            OPERAND OF A GDULIST MACRO. CODING SUPPRES=NO
*                            IN ANY OF THE FOLLOWING DISPGUID MACROS
*                            INDICATES THAT WHENEVER THE DEFINED DATA IN
*                            THE PARTICULAR MACRO IS WRITTEN TO THE 2798
*                            DISPLAY BY THE 2715, THAT DATA WILL BE RETURNED
*                            TO THE 2715 ON THE NEXT ACTIVATION OF THE
*                            ENTER KEY UNLESS IT HAS BEEN CHANGED BY THE
*                            OPERATOR. CODING SUPPRES=YES OR OMITTING THE
*                            OPERAND INDICATES THAT WHENEVER THE DEFINED
*                            DATA IN THE PARTICULAR DISPGUID MACRO IS
*                            WRITTEN TO THE 2798 DISPLAY BY THE 2715, THAT
*                            DATA WILL NOT BE RETURNED TO THE 2715 ON THE
*                            NEXT ACTIVATION OF THE ENTER KEY.
*
*
DG1        DISPGUID DISPMSG='2=1ENTR TESTDATA'
DG2        DISPGUID DISPMSG='DEPRESS ENTER',SUPPRES=NO
DG3        DISPGUID DISPMSG='STEP 2'
DG4        DISPGUID DISPMSG='STEP 3'
DG5        DISPGUID DISPMSG='STEP 4'
DG6        DISPGUID DISPMSG='STEP 5'
DG7        DISPGUID DISPMSG='GET/STORE'
DG8        DISPGUID DISPMSG='3-1ENTR SERVCODE'
DG9        DISPGUID DISPMSG='3-2BLDG/COLUMN'
DG10       DISPGUID DISPMSG='MAT 1-1'
DG11       DISPGUID DISPMSG='MAT 2-2'
DG12       DISPGUID DISPMSG='SELECT LEVR TO 3'
DG14       DISPGUID DISPMSG='MAT 1-2      SL',SUPPRES=NO
DG15       DISPGUID DISPMSG='OLD PART'
DG16       DISPGUID DISPMSG='NEW PART'
DG17       DISPGUID DISPMSG='TRANSLATE'
DG18       DISPGUID DISPMSG='ENTER TEXT'
DG19       DISPGUID DISPMSG='LOCATE20-ORDER21'
DG20       DISPGUID DISPMSG='STOCK24-INPROC25'
DG21       DISPGUID DISPMSG='PRICE22-OTHER23'
DG22       DISPGUID DISPMSG='QUO35-LP36-QTY37'
DG23       DISPGUID DISPMSG='LT26-RAT27-SUP28'
DG24       DISPGUID DISPMSG='IT29-LIN30-BIN31'
DG25       DISPGUID DISPMSG='RM32-ORD33-QC34'
DG26       DISPGUID DISPMSG='WAIT FOR ANSWER'
DG27       DISPGUID DISPMSG='239511',SUPPRES=NO
```

```
DG28     DISPGUID DISPMSG='TOTAL PURCHASE'
DG29     DISPGUID DISPMSG='3-2 TO ADR='
DG37     DISPGUID DISPMSG='QUAD EQN A='
DG38     DISPGUID DISPMSG='B='
DG39     DISPGUID DISPMSG='C='
*
*
*                       THE FOLLOWING TRANSLAT MACROS EACH ASSOCIATE
*                       A USER DEFINED TRANSLATE CHARACTER WITH UP TO
*                       14 CHARACTERS OF TEXT.  THE USER CAN ONLY USE
*                       THE TRANSLATE FUNCTION ON ANY TRANSACTION
*                       STEP (GDULIST MACRO) THAT HAS A PARAMETER LIST
*                       NUMBER (PARAMNO OPERAND) ASSOCIATED WITH A
*                       PARMLIST MACRO THAT HAS TRANSL=YES CODED.
*
*
         TRANSLAT TRANSCH=C3,TRANTXT='CE'
         TRANSLAT TRANSCH=C4,TRANTXT='DOCTOR'
         TRANSLAT TRANSCH=C6,TRANTXT='FIRE'
         TRANSLAT TRANSCH=C9,TRANTXT='IBM MAINT'
         TRANSLAT TRANSCH=D4,TRANTXT='MOVER REQUIRED'
         TRANSLAT TRANSCH=D9,TRANTXT=';N'
         TRANSLAT TRANSCH=E3,TRANTXT='TEL REPAIR'
         TRANSLAT TRANSCH=E5,TRANTXT='VENDING MACH'
*                       THE FOLLOWING MACROS DEFINE THE USER
*                       TRANSACTIONS. EACH TRANSACTION BEGINS WITH A
*                       TRLIST MACRO WHICH GENERATES THE TRANSACTION
*                       LIST HEADER AND CONTAINS FROM 1 TO 16
*                       MACROS: ASLIST, DEULIST, CTRLIST, GDULIST. FROM 1
*                       TO 160 TRANSACTIONS MAY BE SPECIFIED BY THE
*                       USER WITH TRID VALUES BETWEEN 0 AND 159.
*
*
CPU      TRLIST ROUTE=CPU,TRID=0
         DEULIST
*                       THE CPU TRANSACTION CONSISTS OF 1 STEP AND
*                       WILL BE ROUTED TO THE CPU.
BADGE    TRLIST ROUTE=(LOG),TRID=1
         ASLIST B,NORM=19
*                       THE BADGE TRANSACTION CONSISTS OF 1 STEP AND
*                       WILL BE ROUTED TO THE 2740 ATTACHED TO THE 2715.
*                       THE DATA ENTRY WILL BE A BADGE ENTERED AT THE
*                       AREA STATION WITH GUIDANCE LIGHT 19 TURNED ON
*                       WHEN THE TRANSACTION STEP IS ENTERED.
BADGE1   TRLIST ROUTE=(CPU,LOG),TRID=2,DEMOD10=YES,INQDISP=YES
         ASLIST B,NORM=31,MODULUS=(2,10,4),LENGTH=(11,2),INQDISP=7
*                       THE BADGE1 TRANSACTION CONSISTS OF 1 STEP AND
*                       WILL BE ROUTED TO BOTH THE CPU AND THE 2740. A
*                       MODULUS 10 CHECK WILL BE PERFORMED ON POSITIONS
```

```
*                             2 THROUGH 10 AND WILL BE CHECKED WITH THE SELF-
*                             CHECK CHARACTER IN POSITION 11. IF THE MODULUS
*                             10 CHECK FAILS, GUIDANCE LIGHT 4 WILL BE TURNED
*                             ON. THE DATA ENTRY WILL BE A BADGE ENTERED AT
*                             THE AREA STATION WITH GUIDANCE LIGHT 4 TURNED
*                             ON WHEN THE TRANSACTION STEP IS ENTERED. THE
*                             LENGTH OF THE DATA ENTRY WILL ALSO BE CHECKED
*                             AND IF THE LENGTH IS NOT 11, GUIDANCE LIGHT 2
*                             WILL BE TURNED ON. THIS TRANSACTION IS ALSO AN
*                             INQUIRY DISPLAY TRANSACTION. GUIDANCE LIGHT 7 ON
*                             THE AREA STATION WILL BE TURNED ON WHEN THIS
*                             TRANSACTION IS RECEIVED BY THE 2715 AND ROUTED
*                             TO THE CPU AS PRIORITY DATA. THIS IS REALLY THE
*                             INQUIRY-IN-PROCESS GUIDANCE LIGHT. IF THE
*                             INQUIRY IS ABORTED BY THE OPERATOR AT THE 2791
*                             AREA STATION, GUIDANCE LIGHT 1 WILL BE TURNED ON
*                             AUTOMATICALLY. ALL AREA STATIONS THAT USE
*                             INQUIRY DISPLAY TRANSACTIONS MUST RESERVE
*                             GUIDANCE LIGHT 1 FOR THE INQUIRY ABORT
*                             SITUATION.
MANUAL   TRLIST ROUTE=(DISK,LOG),TRID=3
         ASLIST M,NORM=27,LENGTH=(5,23)
         ASLIST M,NORM=26,LENGTH=(5,22)
         ASLIST M,NORM=25,LENGTH=(7,21),ENTRY=M
*                             THE MANUAL TRANSACTION CONSISTS OF 3 STEPS
*                             AND WILL BE ROUTED TO THE 2715 DISK AS DEFERRED
*                             DATA AND TO THE 2740.  MANUAL DATA ENTRIES WILL
*                             BE MADE FOR ALL 3 STEPS. FOR THE FIRST STEP,
*                             GUIDANCE LIGHT 27 WILL BE TURNED ON WHEN THE
*                             STEP IS ENTERED AND GUIDANCE LIGHT 23 WILL BE
*                             TURNED ON IF THE DATA ENTRY LENGTH IS NOT 5.
*                             FOR THE SECOND STEP, GUIDANCE LIGHT 26 WILL BE
*                             TURNED ON WHEN THE STEP IS ENTERED AND GUIDANCE
*                             LIGHT 22 WILL BE TURNED ON IF THE DATA ENTRY
*                             LENGTH IS NOT 5. FOR THE THIRD STEP, GUIDANCE
*                             LIGHT 25 WILL BE TURNED ON WHEN THE STEP IS
*                             ENTERED AND GUIDANCE LIGHT 21 WILL BE TURNED ON
*                             IF THE DATA LENGTH IS NOT 7. THE THIRD STEP IS
*                             A MULTIPLE ENTRY STEP SO THAT 7 CHARACTERS CAN
*                             BE ENTERED.
CARD     TRLIST ROUTE=(LOG),TRID=4
         ASLIST C,NORM=17
CARD1    TRLIST ROUTE=(DISK,LOG),TRID=5
         ASLIST C,NORM=30,DIGIT=(2,1,10),LENGTH=(47,11)
         ASLIST M,NORM=18,LENGTH=(6,11)
*                             THE CARD1 TRANSACTION CONSISTS OF 2 STEPS AND
*                             WILL BE ROUTED TO THE 2715 DISK AND TO THE
```

```
*                             2740. THE FIRST STEP WILL BE A CARD ENTRY WITH
*                             GUIDANCE LIGHT 30 TURNED ON WHEN THE STEP IS
*                             ENTERED, IF THE CHARACTER IN POSITION 2 OF THE
*                             DATA ENTRY IS NOT THE EBCDIC CHARACTER F1, THEN
*                             GUIDANCE LIGHT 10 IS TURNED ON. IF THE LENGTH OF
*                             THE CARD ENTRY IS NOT 47, THEN GUIDANCE LIGHT 11
*                             IS TURNED ON. THE SECOND STEP WILL BE A MANUAL
*                             ENTRY WITH GUIDANCE LIGHT 18 TURNED ON WHEN THE
*                             STEP IS ENTERED. IF THE LENGTH IS NOT 6, THEN
*                             GUIDANCE LIGHT 11 IS TURNED ON.
CARD2    TRLIST ROUTE=(CPU,LOG),TRID=6,DEMOD11=YES,INQDISP=YES
         ASLIST C,NORM=29,MODULUS=(2,15,5),LENGTH=(17,11)
         ASLIST B,NORM=20,LENGTH=(11,2)
         ASLIST M,NORM=15,INQDISP=7
EXP1     TRLIST ROUTE=(CPU,LOG),TRID=7,DEMOD10=YES,INQDISP=YES
         ASLIST B,NORM=16,MODULUS=(2,9,4),INQDISP=7
EXP2     TRLIST ROUTE=(CPU,LOG),TRID=8,DEMOD11=YES
         ASLIST B,NORM=16,MODULUS=(2,9,5)
*                             THE EXP2 TRANSACTION CONSISTS OF 1 STEP AND
*                             WILL BE ROUTED TO THE CPU AND TO THE 2740. THE
*                             DATA ENTRY WILL BE A BADGE ENTERED AT THE AREA
*                             STATION WITH GUIDANCE LIGHT 16 TURNED ON WHEN
*                             THE TRANSACTION STEP IS ENTERED. A MODULUS 11
*                             CHECK WILL BE PERFORMED ON POSITIONS 2 THROUGH
*                             10 AND WILL BE CHECKED WITH THE SELF-CHECK
*                             CHARACTER IN POSITION 11. IF THE MODULUS 11
*                             CHECK FAILS, GUIDANCE LIGHT 5 WILL BE TURNED ON.
EXP3     TRLIST ROUTE=(CPU,LOG),TRID=9,DEMOD10=YES,INQDISP=YES
         ASLIST B,NORM=16,MODULUS=(2,7,4)
         ASLIST C,NORM=17,INQDISP=7,ENTRY=M
EXP4     TRLIST ROUTE=(LOG),TRID=10
         ASLIST B,NORM=16,DIGIT=(3,5,10),LENGTH=(11,11)
*                             THE EXP4 TRANSACTION CONSISTS OF 1 STEP AND
*                             WILL BE ROUTED TO THE 2740. THE STEP WILL BE A
*                             BADGE ENTRY WITH GUIDANCE LIGHT 16 TURNED ON
*                             WHEN THE STEP IS ENTERED. GUIDANCE LIGHT 10 WILL
*                             BE TURNED ON BY THE 2715 IF THE CHARACTER IN
*                             POSITION 3 IS NOT THE EBCDIC CHARACTER F5.
*                             GUIDANCE LIGHT 11 WILL BE TURNED ON IF THE
*                             LENGTH OF THE DATA ENTRY IS NOT 11.
EXP5     TRLIST ROUTE=(LOG),TRID=11
         ASLIST B,NORM=16
DEU1     TRLIST ROUTE=LOG,TRID=12
         DEULIST DIGIT=(2,1),DIGIT2=(3,1)
*                             THE DEU1 TRANSACTION CONSISTS OF 1 STEP AND
*                             WILL BE ROUTED TO THE 2740. THE DATA ENTRY WILL
*                             BE MADE FROM A DATA ENTRY UNIT. AN ERROR WILL BE
*                             INDICATED AT THE DEU IF POSITION 2 DOES NOT
*                             CONTAIN THE EBCDIC CHARACTER F1 OR IF POSITION
*                             3 DOES NOT CONTAIN THE EBCDIC CHARACTER F1.
```

```
DEU2      TRLIST ROUTE=(LOG),TRID=13,DEMOD10=YES
          DEULIST DIGIT2=(2,1),MODULUS=(3,10)
DEU3      TRLIST ROUTE=(LOG),TRID=14,DEMOD11=YES
          DEULIST DIGIT2=(2,5),MODULUS=(3,10),LENGTH=13
DEU4      TRLIST ROUTE=(LOG),TRID=15
          DEULIST DIGIT=(2,6),DIGIT2=(3,9) LENGTH=11
DEUEXP    TRLIST ROUTE=(LOG,NULL),DEMOD10=YES,TRID=16,TEXT=YES
          DEULIST MODULUS=(3,10),MSG=' THIS IS AN EXPANDED TRANSACTION'
*                        THE DEUEXP TRANSACTION CONSISTS OF 1 STEP AND
*                        WILL BE ROUTED TO THE 2740 AND TO THE PRINTER
*                        ATTACHED TO THE AREA STATION THAT WILL BE
*                        SPECIFIED BY THE OPERATOR IN THE FIRST DATA
*                        ENTRY. THE DATA ENTRY WILL BE ENTERED FROM A DEU
*                        AND A DEFINED MESSAGE (IMPLICIT TEXT) WILL BE
*                        INCLUDED WITH THE TRANSACTION. A MODULUS 10
*                        CHECK WILL BE PERFORMED ON POSITIONS 3 THROUGH
*                        10 AND WILL BE COMPARED WITH THE CHECK
*                        CHARACTER IN POSITION 11. IF THE MODULUS 10
*                        CHECK FAILS, THEN THE RED ERROR BUTTON WILL BE
*                        INDICATED AT THE DEU.
DEU6      TRLIST ROUTE=(LOG),TRID=17
          DEULIST
EXPALM    TRLIST ROUTE=(CPU,59),TEXT=YES,TRID=18
          DEULIST MSG='' VENI VIDI VICI AT TWO PRINTERS, I HOPE'
*                        THE EXPALM TRANSACTION CONSISTS OF 1 STEP AND
*                        WILL BE ROUTED TO THE CPU AND TO THE PRINTER
*                        ON THE AREA STATION WHOSE ID IS 59. THE DATA
*                        ENTRY WILL BE ENTERED FROM A DEU AND A DEFINED
*                        MESSAGE WILL BE ROUTED ALONG WITH THE
*                        TRANSACTION.
EXPALMTX  TRLIST ROUTE=(CPU,59),TEXT=YES,TRID=19
          DEULIST MSG=' ' TYPE AT TWO PRINTER AND NO ALARM'
EXPTX     TRLIST ROUTE=42,TRID=20
          DEULIST
          CTRLIST DEVCOD=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,            X
                CTROP=READ
RDIPSG    TRLIST ROUTE=LOG,TRID=21
          DEULIST
          CTRLIST DEVCOD=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,            X
                CTROP=READ
*                        THE RDIPSG TRANSACTION CONSISTS OF 2 STEPS AND
*                        WILL BE ROUTED TO THE 2740. THE FIRST STEP IS A
*                        DATA ENTRY FROM A DEU. THE SECOND STEP IS THE
*                        PULSE COUNT DATA ENTRY. THIS STEP WILL CAUSE THE
*                        SINGLE COUNTER WHOSE IMPLIED ADDRESS RESULTS
*                        FROM THE CONVERSION OF THE DEVICE ADDRESS OF
*                        THE DEU INITIATING THE REQUEST. THERE WILL BE
*                        NO CHANGE IN THE PRESENT COUNT TEST CONDITION
*                        OF THE COUNTER.
```

```
RDEPSGM         TRLIST   ROUTE=LOG,TRID=22
                DEULIST
                CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,     X
                      CTROP=READ
*                         THE RDEPSGM TRANSACTION CONSISTS OF 2 STEPS
*                         AND WILL BE ROUTED TO THE 2740. THE FIRST STEP
*                         IS A DATA ENTRY FROM A DEU. THE SECOND STEP IS
*                         THE PULSE COUNT DATA ENTRY SET UP TO READ THE
*                         SINGLE COUNTER WHOSE ADDRESS IS EXPLICITLY
*                         SPECIFIED IN THE MANUAL DATA ENTRY. THERE WILL
*                         BE NO CHANGE IN THE PRESENT COUNT TEST
*                         CONDITION OF THE COUNTER.
RDEPSGB         TRLIST ROUTE=LOG,TRID=23
                DEULIST
                CTRLIST DEVCOD=B,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,     X
                      CTROP=READ
RDEPGPM         TRLIST ROUTE=LOG,TRID=24
                DEULIST
                CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=GROUP,CTTEST=NULL,      X
                      CTROP=READ
RDEPGPB         TRLIST ROUTE=LOG,TRID=25
                DEULIST
                CTRLIST DEVCOD=B,CTRADR=EXP,CTRRD=GROUP,CTTEST=NULL,      X
                      CTROP=READ
*                         THE RDEPGPB TRANSACTION CONSISTS OF 2 STEPS
*                         AND WILL BE ROUTED TO THE 2740. THE FIRST STEP
*                         IS A DATA ENTRY FROM A DEU. THE SECOND STEP IS
*                         THE PULSE COUNT DATA ENTRY SET UP TO READ THE
*                         GROUP OF COUNTERS THAT WILL BE EXPLICITLY
*                         SPECIFIED IN THE BADGE DATA ENTRY. THERE WILL
*                         BE NO CHANGE IN THE PRESENT COUNT TEST
*                         CONDITIONS OF ANY OF THE COUNTERS.
RDRSTIP         TRLIST ROUTE=LOG,TRID=26
                DEULIST
                CTRLIST DEVCOD=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,     X
                      CTROP=READRST
*                         THE RDRSTIP TRANSACTION CONSISTS OF 2 STEPS
*                         AND WILL BE ROUTED TO THE 2740. THE FIRST STEP
*                         IS A DATA ENTRY FROM A DEU. THE SECOND STEP IS
*                         THE PULSE COUNT DATA ENTRY. THE SINGLE COUNTER,
*                         WHOSE ADDRESS IS IMPLIED FROM THE CONVERSION
*                         OF THE DEVICE ADDRESS OF THE DEU INITIATING
*                         THE REQUEST, WILL BE READ AND THEN THAT COUNTER
*                         WILL BE RESET. THERE WILL BE NO CHANGE IN THE
*                         PRESENT COUNT TEST CONDITION OF THE COUNTER.
```

```
RDRSTEPM        TRLIST ROUTE=LOG,TRID=27
                DEULIST
                CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,      X
                      CTROP=READRST
*** FIRST SET OF EXPANSION TRANSACTIONS FOR 2796 TC81-TC89 ***
*********                                            *********
RDSTIPM         TRLIST ROUTE=(LOG,42),TRID=28
                DEULIST
                CTRLIST DEVCOD=M,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,      X
                      CTROP=READSET
RDSTEPM         TRLIST ROUTE=(LOG,42),TRID=29
                DEULIST
                CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,      X
                      CTROP=READSET
*                         THE RDSTEPM TRANSACTION CONSISTS OF 2 STEPS
*                         AND WILL BE ROUTED TO THE 2740 AND TO THE
*                         PRINTER ON THE AREA STATION WHOSE ID IS 42.
*                         THE FIRST STEP IS A DATA ENTRY FROM A DEU.
*                         THE SECOND STEP IS THE PULSE COUNT DATA ENTRY
*                         SET UP TO READ THE SINGLE COUNTER, WHOSE ADDRESS
*                         IS EXPLICITLY SPECIFIED IN THE MANUAL DATA
*                         ENTRY, AND THEN TO SET THE COUNTER TO THE
*                         EXPLICITLY SPECIFIED VALUE. THERE WILL BE NO
*                         CHANGE IN THE PRESENT COUNT TEST CONDITION OF
*                         THE COUNTER.
RDSTEPB         TRLIST ROUTE=(LOG,42),TRID=30
                DEULIST
                CTRLIST DEVCOD=B,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,      X
                      CTROP=READSET
RDSIDEPM        TRLIST ROUTE=(LOG,42),TRID=31
                DEULIST
                CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,      X
                      CTROP=RDRESID
RDSIDGPM        TRLIST ROUTE=(LOG,42),TRID=32
                DEULIST
                CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=GROUP,CTTEST=NULL,       X
                      CTROP=RDRESID
RDSIDGPB        TRLIST ROUTE=(LOG,42),TRID=33
                DEULIST
                CTRLIST DEVCOD=B,CTRADR=EXP,CTRRD=GROUP,CTTEST=NULL,       X
                      CTROP=RDRESID
WRIPB           TRLIST ROUTE=LOG,TRID=34
                DEULIST
                CTRLIST DEVCOD=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,      X
                      CTROP=SET
```

```
WREPM           TRLIST ROUTE=LOG,TRID=35
                DEULIST
                CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,    X
                      CTROP=SET
*                         THE WREPM TRANSACTION CONSISTS OF 2 STEPS AND
*                         WILL BE ROUTED TO THE 2740. THE FIRST STEP IS
*                         THE PULSE COUNT DATA ENTRY SET UP TO SET THE
*                         SINGLE COUNTER, WHOSE ADDRESS IS EXPLICITLY
*                         SPECIFIED IN THE MANUAL ENTRY, TO THE MANUAL
*                         VALUE SPECIFIED AT THE DEU. THERE WILL BE NO
*                         CHANGE IN THE PRESENT COUNT TEST CONDITION OF
*                         THE COUNTER.
RDGPEPAA        TRLIST ROUTE=42,TRID=36
                DEULIST
                CTRLIST DEVCOD=B,CTRADR=EXP,CTRRD=GROUP,CTTEST=NULL,     X
                      CTROP=READ
RDGPEPBB        TRLIST ROUTE=59,TRID=37
                DEULIST
                CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=GROUP,CTTEST=NULL,     X
                CTROP=READ
RDSIEPAA        TRLIST ROUTE=42,TEXT=YES,TRID=38
                DEULIST
                CTRLIST DEVCOD=B,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,    X
                      CTROP=READ,MSG='A SINGLE COUNTER SHOULD ACCOMPANY  X
                      THIS MESSAGE'
RDMSIEP         TRLIST ROUTE=42,TEXT=YES,TRID=39
                DEULIST
                CTRLIST DEVCOD=M,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,    X
                      CTROP=READ,MSG='A SINGLE COUNTER SHOULD ACCOMPANY  X
                      THIS MESSAGE'
RDSIIPAA        TRLIST ROUTE=42,TEXT=YES,TRID=40
                DEULIST
                CTRLIST DEVCOD=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,    X
                      CTROP=READ,MSG=''''THIS IS AN ALARM MESSAGE WITH   X
                      A COUNTER'
RDSIIPAB        TRLIST ROUTE=59,TEXT=YES,TRID=41
                DEULIST
                CTRLIST DEVCOD=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,    X
                   CTROP=READ,MSG-''''ALARM MESSAGE AT TWO AREA STATIONS X
                   AND A COUNTER VALUE AT ONE'
EXP             TRLIST ROUTE=LOG,TRID=42
                DEULIST
                CTRLIST DEVCOD=B,CTRADR=EXP,CTRRD=SINGLE,CTTEST=NULL,    X
                      CTROP=READ
READ            TRLIST ROUTE=LOG,TRID=43
                DEULIST
                CTRLIST DEVCOD=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,    X
                      CTROP=READ
```

```
READST          TRLIST ROUTE=LOG,TRID=44
                DEULIST
                CTRLIST DEVCOD=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,      X
                      CTROP=READSET
READSID         TRLIST ROUTE=LOG,TRID=45
                DEULIST
                CTRLIST DEVCOD=B,CTRADR=IMP,CTRRD=SINGLE,CTTEST=NULL,      X
                      CTROP=RDRESID
ALARM           TRLIST ROUTE=(LOG,42),TEXT=YES,TRID=46
                DEULIST MSG='' '
ALARMTX         TRLIST ROUTE=(LOG,42),TEXT=YES,TRID=47
                DEULIST MSG=''' VENI VIDI VICI'
TEXT            TRLIST ROUTE=(LOG,42),TEXT=YES,TRID=48
                DEULIST MSG=' ' THE ALARM HAD BETTER NOT HAVE SOUNDED'
DISK            TRLIST ROUTE=DISK,TRID=49
                DEULIST
TESTO           TRLIST TRID=60,ROUTE=LOG
                GDULIST PARAMNO=01,NORGUID=1,DISPMSG=DG17
                GDULIST PARAMNO=02,NORGUID=1,DISPMSG=DG1
                GDULIST PARAMNO=33,NORGUID=(1,3,5),IDENT=4
*                         THE TESTO TRANSACTION CONSISTS OF 3 STEPS AND
*                         WILL BE ROUTED TO THE 2740. ALL 3 STEPS WILL
*                         BE DATA ENTRIES FROM THE 2798. WHEN THE FIRST IS
*                         ENTERED, GUIDANCE LIGHT 1 IS TURNED ON AT THE
*                         OPERATOR GUIDANCE PANEL AND THE MESSAGE DEFINED
*                         BY DISPGUID MACRO DG17 WILL BE DISPLAYED ON THE
*                         2798 GUIDANCE DISPLAY PANEL. THE 2715 WILL USE
*                         PARAMETER LIST NUMBER 01 TO GET TO THE
*                         PARAMETER LIST DEFINED BY PARMLIST MACRO, PAR1.
*                         THIS PARAMETER LIST WILL BE USED IN CHECKING
*                         THE FIRST DATA ENTRY. WHEN THE SECOND STEP IS
*                         ENTERED, GUIDANCE LIGHT 1 IS TURNED ON AT THE
*                         OPERATOR GUIDANCE PANEL AND THE MESSAGE DEFINED
*                         BY DISPGUID MACRO DG1 WILL BE DISPLAYED ON THE
*                         2798 GUIDANCE DISPLAY PANEL. THE 2715 WILL USE
*                         PARAMETER LIST NUMBER 02 TO GET TO THE
*                         PARAMTER LIST DEFINED BY THE PARMLIST MACRO
*                         PAR2. THIS PARAMETER LIST WILL BE USED IN
*                         CHECKING THE SECOND DATA ENTRY. WHEN THE THIRD
*                         STEP IS ENTERED, GUIDANCE LIGHTS 1,3,AND 5 WILL
*                         BE TURNED ON AT THE OPERATOR GUIDANCE PANEL AND
*                         THE MESSAGE DEFINED IN THE FIFTH IDENTIFIER
*                         IN THE IDENTIFIER TABLE WILL BE DISPLAYED ON
*                         THE 2798 GUIDANCE DISPLAY PANEL. THE 2715 WILL
*                         USE PARAMETER LIST NUMBER 33 TO GET TO THE
*                         PARAMETER LIST DEFINED BY THE PARMLIST MACRO
*                         PAR33. THIS PARAMETER LIST WILL BE USED IN
*                         CHECKING THE THIRD DATA ENTRY.
```

```
TEST1           TRLIST TRID=61,ROUTE=LOG
                GDULIST PARAMNO=03,NORGUID=1,DISPMSG=DG1
                GDULIST PARAMNO=04,NORGUID=1,DISPMSG=DG3
TEST2           TRLIST TRID=62,ROUTE=LOG
                GDULIST PARAMNO=05,NORGUID=1,DISPMSG=DG1
                GDULIST PARAMNO=06,NORGUID=1,DISPMSG=DG3
                GDULIST PARAMNO=07,NORGUID=1,DISPMSG=DG4
                GDULIST PARAMNO=08,NORGUID=1,DISPMSG=DG5
                GDULIST PARAMNO=09,NORGUID=1,DISPMSG=DG6
TEST3           TRLIST TRID=63,ROUTE=LOG
                GDULIST PARAMNO=10,NORGUID=1,DISPMSG=DG1
                GDULIST PARAMNO=11,NORGUID=1,DISPMSG=DG3
                GDULIST PARAMNO=12,NORGUID=1,DISPMSG=DG4
                GDULIST PARAMNO=13,NORGUID=1,DISPMSG=DG5
                GDULIST PARAMNO=14,NORGUID=1,DISPMSG=DG6
TEST4           TRLIST TRID=64,ROUTE=LOG
                GDULIST PARAMNO=15,NORGUID=1,DISPMSG=DG1
                GDULIST PARAMNO=16,NORGUID=1,DISPMSG=DG3
                GDULIST PARAMNO=17,NORGUID=1,DISPMSG=DG4
TEST5           TRLIST TRID=65,ROUTE=LOG
                GDULIST PARAMNO=18,NORGUID=1,DISPMSG=DG1
                GDULIST PARAMNO=19,NORGUID=1,DISPMSG=DG3
                GDULIST PARAMNO=20,NORGUID=1,DISPMSG=DG4
                GDULIST PARAMNO=21,NORGUID=1,DISPMSG=DG5
TEST6           TRLIST TRID=66,ROUTE=LOG
                GDULIST PARAMNO=22,NORGUID=1,DISPMSG=DG1
                GDULIST PARAMNO=23,NORGUID=1,DISPMSG=DG3
TEST7           TRLIST TRID=67,ROUTE=LOG
                GDULIST PARAMNO=24,NORGUID=1,DISPMSG=DG11
                GDULIST PARAMNO=38,NORGUID=1,DISPMSG=DG3
TEST8           TRLIST TRID=68,ROUTE=LOG
                GDULIST PARAMNO=26,NORGUID=1,DISPMSG=DG1
                GDULIST PARAMNO=37,NORGUID=1,DISPMSG=DG7
ROUTE1          TRLIST TRID=70,ROUTE=(LOG,42),TEXT=YES
                GDULIST PARAMNO=28,NORGUID=(1,5),DISPMSG=DG9
                GDULIST PARAMNO=02,NORGUID-1,DISPMSG=DG8,ENTRY=M,             *
                      MSG='**EMERGENCY'
*                         THE ROUTE1 TRANSACTION CONSISTS OF 2 STEPS AND
*                         WILL BE ROUTED TO THE 2740 AND TO THE PRINTER
*                         ON THE AREA STATION WHOSE ID IS 42. BOTH STEPS
*                         WILL BE DATA ENTRIES FROM THE 2798. WHEN THE
*                         FIRST STEP IS ENTERED, GUIDANCE LIGHTS 1 AND 5
*                         ARE TURNED ON AT THE OPERATOR GUIDANCE PANEL
*                         AND THE MESSAGE DEFINED BY DISPGUID MACRO DG9
*                         WILL BE DISPLAYED ON THE 2798 GUIDANCE DISPLAY
*                         PANEL. THE 2715 WILL USE PARAMETER LIST NUMBER
*                         28 TO GET TO THE PARAMETER LIST DEFINED BY THE
*                         PARMLIST MACRO PAR28. THIS PARAMETER LIST WILL
*                         BE USED BY THE 2715 IN CHECKING THE FIRST DATA
```

```
*                    ENTRY. WHEN THE SECOND STEP IS ENTERED, GUIDANCE
*                    LIGHT 1 IS TURNED ON AT THE OPERATOR GUIDANCE
*                    PANEL AND THE MESSAGE DEFINED BY DISPGUID MACRO
*                    DG8 WILL BE DISPLAYED ON THE 2798 GUIDANCE
*                    DISPLAY PANEL. THE 2715 WILL USE PARAMETER LIST
*                    NUMBER 02 TO GET TO THE PARAMETER LIST DEFINED
*                    BY THE PARMLIST MACRO PAR2. THIS PARAMETER LIST
*                    WILL BE USED BY THE 2715 IN CHECKING THE
*                    SECOND DATA ENTRY. MULTIPLE ENTRIES CAN BE
*                    ENTERED ON THE SECOND STEP. IMPLICIT TEXT WILL
*                    BE INCLUDED WITH THE TRANSACTION WHEN IT IS
*                    ROUTED.
ROUTE2         TRLIST TRID=71,ROUTE=(LOG,NULL)
               GDULIST PARAMNO=38,NORGUID=1,DISPMSG=DG29
               GDULIST PARAMNO=28,NORGUID=(1,5),DISPMSG=DG9
               GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG18
TESTJOB1       TRLIST TRID=72,ROUTE=LOG
               GDULIST PARAMNO=29,NORGUID=(1,2),DISPMSG=DG11
               GDULIST PARAMNO=33,NORGUID=4
               GDULIST PARAMNO=31,NORGUID=5
TESTJOB2       TRLIST TRID=73,ROUTE=LOG
               GDULIST PARAMNO=29,NORGUID=(1,2),DISPMSG=DG11
               GDULIST PARAMNO=32,NORGUID=3
               GDULIST PARAMNO=33,NORGUID=6
               GDULIST PARAMNO=33,NORGUID=(1,7),DISPMSG=DG11
               GDULIST PARAMNO=33,NORGUID=(1,8),DISPMSG=DG11
TESTJOB3       TRLIST TRID=74,ROUTE=LOG
               GDULIST PARAMNO=29,NORGUID=(1,2),DISPMSG=DG11
               GDULIST PARAMNO=33,NORGUID=4
               GDULIST PARAMNO=34,NORGUID=5
CARDORD        TRLIST TRID=75,ROUTE=LOG
               GDULIST PARAMNO=35,NORGUID=(1,4),DISPMSG=DG14
               GDULIST PARAMNO=33,NORGUID=7
UALMAINT       TRLIST TRID=76,ROUTE=LOG
               GDULIST PARAMNO=33,NORGUID=(1,2),DISPMSG=DG10
               GDULIST PARAMNO=33,NORGUID=3,IDENT=4
               GDULIST PARAMNO=36,NORGUID=4
               GDULIST PARAMNO=33,NORGUID=(1,5),DISPMSG=DG15
               GDULIST PARAMNO=33,NORGUID=(1,5),DISPMSG=DG16
INV1           TRLIST TRID=79,ROUTE=LOG
               GDULIST PARAMNO=20,NORGUID=(1,8),DISPMSG=DG19
INV2           TRLIST TRID=80,ROUTE=LOG
               GDULIST PARAMNO=20,NORGUID=(1,8),DISPMSG=DG20
INV3           TRLIST TRID=81,ROUTE=LOG
               GDULIST PARAMNO=20,NORGUID=(1,8),DISPMSG=DG21
INV4           TRLIST TRID=82,ROUTE=LOG
               GDULIST PARAMNO=20,NORGUID=(1,8),DISPMSG=DG22
INV5           TRLIST TRID=83,ROUTE=LOG
               GDULIST PARAMNO=20,NORGUID=(1,8),DISPMSG=DG23
```

```
STOCK           TRLIST TRID=84,ROUTE=CPU,INQDISP=YES
                GDULIST PARAMNO=39,NORGUID=(1,4,8),DISPMSG=DG24
                GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
*                         IF THE INQUIRY IS ABORTED BY THE OPERATOR AT
*                         THE 2798, GDU GUIDANCE LIGHT 16 WILL BE TURNED
*                         ON AUTOMATICALLY AT THE 2798. ALL 2798'S THAT
*                         USE INQUIRY DISPLAY TRANSACTIONS MUST RESERVE
*                         GUIDANCE LIGHT 16 FOR THE INQUIRY ABORT
*                         SITUATION.
INPROC          TRLIST TRID=85,ROUTE=CPU,INQDISP=YES
                GDULIST PARAMNO=39,NORGUID=(1,4,8),DISPMSG=DG25
                GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
LEADTIME        TRLIST TRID=86,ROUTE=CPU,INQDISP=YES
                GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
                GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
RATING          TRLIST TRID=87,ROUTE=CPU,INQDISP=YES
                GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
                GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
SUPPLIER        TRLIST TRID=88,ROUTE=CPU,INQDISP=YES
                GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
                GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
INTRANS         TRLIST TRID=89,ROUTE=CPU,INQDISP=YES
                GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
                GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
LINE            TRLIST TRID=90,ROUTE=CPU,INQDISP=YES
                GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
                GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
BIN             TRLIST TRID=91,ROUTE=CPU,INQDISP=YES
                GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
                GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
RAWMAT          TRLIST TRID=92,ROUTE=CPU,INQDISP=YES
                GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
                GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
ORDER           TRLIST TRID=93,ROUTE=CPU,INQDISP=YES
                GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
                GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
QUALCON         TRLIST TRID=94,ROUTE=CPU,INQDISP=YES
                GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
                GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
QUOTE           TRLIST TRID=95,ROUTE=CPU,INQDISP=YES
                GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
                GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
LASTPUR         TRLIST TRID=96,ROUTE=CPU,INQDISP=YES
                GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
                GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
ECONQTY         TRLIST TRID=97,ROUTE=CPU,INQDISP=YES
                GDULIST PARAMNO=39,NORGUID=(1,4),DISPMSG=DG27
                GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
```

```
CREDIT          TRLIST TRID=98,ROUTE=CPU,INQDISP=YES
                GDULIST PARAMNO=27,NORGUID=(1,2),DISPMSG=DG12
                GDULIST PARAMNO=25,NORGUID=1,DISPMSG=DG28
                GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
QUADEQN         TRLIST TRID=99,ROUTE=CPU,INQDISP=YES
                GDULIST PARAMNO=30,NORGUID=1,DISPMSG=DG37
                GDULIST PARAMNO=30,NORGUID=1,DISPMSG=DG38
                GDULIST PARAMNO=30,NORGUID=1,DISPMSG=DG39
                GDULIST PARAMNO=32,NORGUID=1,DISPMSG=DG26,ENTRY=M
                STEND

END
/*
```

```
LOC    OBJECT CODE    ADDR1 ADDR2 STMT   SOURCE STATEMENT                                                   F3OSEP69 10/06/70

000000                                   EXAMPLE2 CSECT
                                         ***********************************************************************
                                         *                                                                     *
                                         *                 SAMPLE EXPANDED ID VERIFICATION PROGRAM             *
                                         *                                                                     *
                                         *  THIS PROGRAM IS DESIGNED FOR A BINARY SYNCHRONOUS SWITCHED LINE    *
                                         *  OVER WHICH AN IBM 2770 OR 2780 CAN CALL THE CENTRAL COMPUTER.      *
                                         *  THE PROGRAM UTILIZES SOME OF THE EXPANDED ID VERIFICATION CAPABILI-*
                                         *  TIES. THE ID OF THE 2780 IS RR: THE ID OF THE 2770 IS WW.          *
                                         *  IF THE USER WISHES TO EXECUTE THIS PROGRAM, HE MUST MODIFY IT      *
                                         *  TO USE THE ID'S OF HIS PARTICULAR TERMINALS.                       *
                                         *                                                                     *
                                         *  OPERATION:                                                         *
                                         *                                                                     *
                                         *  (1.)  WHEN THE PROGRAM IS STARTED, A MESSAGE , 'SWTEST HAS BEGUN   *
                                         *        EXECUTION', WILL BE PRINTED ON THE CONSOLE.                  *
                                         *                                                                     *
                                         *  (2.)  IF THE LINE CAN NOT BE OPENED, A MESSAGE, 'OPEN DID NOT      *
                                         *        COMPLETE SUCCESSFULLY', WILL BE PRINTED ON THE CONSOLE AND   *
                                         *        THE PROGRAM IS ABENDED.                                      *
                                         *                                                                     *
                                         *  (3.)  IF THE LINE IS OPENED SUCCESSFULLY, THE TERMINAL OPERATOR    *
                                         *        CAN THEN DIAL THE COMPUTER FROM EITHER TERMINAL AND          *
                                         *        SEND DATA. WHEN ALL OF THE DATA HAS BEEN READ FROM THE       *
                                         *        TERMINAL, THE PROGRAM DISCONNECTS THE LINE.                  *
                                         *                                                                     *
                                         *  (4.)  THE TERMINAL OPERATOR CAN THE MANUALLY DIAL FROM EITHER      *
                                         *        TERMINAL AND RECEIVE SIX MESSAGES BEFORE THE LINE IS DIS-    *
                                         *        CONNECTED.                                                   *
                                         *                                                                     *
                                         *  (5.) THE PROGRAM THEN ISSUES TWO CHGNTRY MACROS TO CHANGE THE      *
                                         *        CONTROL BYTE VALUE FOR EACH AUTHORIZED ID. A SNAP DUMP IS    *
                                         *        TAKEN BEFORE AND AFTER THE CHGNTRY MACROS IN ORDER TO SHOW   *
                                         *        THE TERMINAL LIST CONTENTS BEFORE AND AFTER ISSUING THE      *
                                         *        CHGNTRY MACROS. (IN ORDER TO RECEIVE THE SNAP                *
                                         *        DUMP, THE USER MUST PROVIDE A DD CARD FOR HIS SNAP DCB DDNAME *
                                         *        IN HIS JCL.)                                                 *
                                         *                                                                     *
                                         *  (6.)  A MESSAGE,  'SWTEST HAS SUCCESSFULLY COMPLETED', WILL BE     *
                                         *        PRINTED ON THE CONSOLE.                                      *
                                         *                                                                     *
                                         *  NOTES -                                                            *
                                         *                                                                     *
                                         *  AFTER EACH READ AND WRITE OPERATION, THE PROGRAM CHECKS FOR        *
                                         *  NORMAL COMPLETION.                                                 *
                                         *                                                                     *
                                         *  IF THE OPERATION WAS A READ -                                      *
                                         *                                                                     *
                                         *     (1.) WHICH COMPLETED NORMALLY WITH NO BITS IN DECFLAGS TURNED   *
                                         *          ON, THE PROGRAM CONTINUES WITH THE NEXT OPERATION.         *
                                         *     (2.) WHICH COMPLETED NORMALLY WITH A BIT ON IN DECFLAGS, THE    *
                                         *          PROGRAM TAKES A SNAP DUMP, THEN CHECKS FOR AN INVALID ID.  *
                                         *                                                                     *
                                         *           (A.) IF AN INVALID ID WAS RECEIVED, A MESSAGE, 'AN INVALID *
                                         *                ID WAS RECEIVED', IS PRINTED ON THE CONSOLE AND THE  *
                                         *                PROGRAM DISCONNECTS THE LINE.                        *
                                         *                                                                     *
                                         *           (B.) IF AN INVALID ID WAS NOT RECEIVED, THE PROGRAM       *
                                         *                DISCONNECTS THE LINE.                                *
                                         *                                                                     *
                                         *     (3.) WHICH ENDED WITH ERROR, THE PROGRAM TAKES A SNAP DUMP AND  *
                                         *           IF A DATA CHECK OCCURRED ON A READ CONTINUE OPERATION,    *
                                         *           PERFORMS A READ REPEAT. IF THE ERROR WAS NOT A DATA CHECK, *
                                         *           AN ERROR MESSAGE, 'ERROR CANNOT BE HANDLED BY PROGRAM', IS *
                                         *           PRINTED ON THE CONSOLE.                                   *
                                         *                                                                     *
                                         *                                                                     *
                                         *  IF THE OPERATION WAS A WRITE -                                     *
                                         *                                                                     *
                                         *     (1.) WHICH COMPLETED NORMALLY WITH NO BITS TURNED ON IN         *
                                         *          DECFLAGS, THE PROGRAM CONTINUES WITH THE NEXT OPERATION.   *
                                         *                                                                     *
                                         *     (2.) WHICH COMPLETED NORMALLY AND IS NOT A WRITE CONNECT, THE   *
                                         *          PROGRAM DISCONNECTS THE LINE.                              *
                                         *                                                                     *
                                         *     (3.) WHICH COMPLETED NORMALLY AND IS A WRITE CONNECT, THE WRITE *
                                         *          CONNECT IS REISSUED IF A WACK OR ID NAK WAS RECEIVED. IF AN *
                                         *          INVALID ID WAS RECEIVED, A MESSAGE, 'INVALID ID RECEIVED - *
                                         *          NOTHING TRANSMITTED TO TERMINAL', IS PRINTED ON THE CONSOLE. *
                                         *          OTHERWISE, THE PROGRAM DISCONNECTS THE LINE.               *
                                         *                                                                     *
                                         *     (4.) WHICH COMPLETED WITH ERROR, THE PROGRAM TAKES A SNAP DUMP  *
                                         *          AND CHECKS FOR A TIMEOUT CONDITION.                        *
                                         *                                                                     *
                                         *           (A.) IF THE ERROR WAS A TIMEOUT, THE PROGRAM REINITIATES  *
                                         *           THE OPERATION.                                            *
                                         *                                                                     *
                                         *           (B.) IF THE ERROR WAS NOT A TIMEOUT,  AN ERROR MESSAGE,   *
                                         *                'ERROR CANNOT BE HANDLED BY PROGRAM', IS PRINTED ON THE *
                                         *                CONSOLE.                                             *
                                         *                                                                     *
                                         ***********************************************************************
                                                  PRINT NOGEN
```

```
                                              95          SAVE  (14,12)                   SAVE REGISTERS

000004 05C0                                   99 ENTRY    BALR  BASEREG,0
000006                                       100          USING *,BASEREG                 ESTABLISH ADDRESSABILITY
000000                                       101          USING IHADCB,DCBREG             ESTABLISH ADDRESSABILITY FOR DCB
000000                                       102          USING IECTDECB,DECBREG            AND DECB

000006 50D0 C7C6                00700        104          ST    SAVEREG,SAVE+4            STORE ADDRESS OF SAVE AREA
00000A 41D0 C7C2                007C8        105          LA    SAVEREG,SAVE
00000E 4160 C6D6                006DC        106          LA    DCBREG,MYDCB
000012 4170 C73A                00740        107          LA    DECBREG,MYDECB

                                             109          WTO   'SWTEST HAS BEGUN EXECUTION'

                                             119          OPEN  (SNAPDCB,(OUTPUT))        OPEN THE SNAP DCB
                                                                                                         PAGE    3

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT                                      F30SEP69  10/06/70

                                             126          OPEN  (MYDCB)                   OPEN THE LINE DCB

000052 9110 6030          00030              133          TM    DCBOFLGS,X'10'            DID OPEN COMPLETE SUCCESSFULLY
000056 4710 C09A                000A0        134          BC    BEGIN                     IF SO, ISSUE READ CONNECT

                                             136          WTO   'OPEN DID NOT COMPLETE SUCCESSFULLY'

000088 47F0 C290                00296        146          B     EXIT


                                             148          PRINT GEN
                                             149 ERRBLOCK LERB  1
00008C                                       150+ERRBLOCK DC    0F'0' ALIGN TABLE AND ATTACH NAME

                                             152+*        ACCUMULATORS
00008C 00000000                              153+         DC    F'0' TRANSMISSIONS
000090 0000                                  154+         DC    H'0' DATA CHECKS
000092 0000                                  155+         DC    H'0' INTERVENTIONS
000094 0000                                  156+         DC    H'0' NONTEXT TIMEOUTS
                                             157+*        COUNTERS
000096 00                                    158+         DC    X'0' TRANSMISSIONS
000097 00                                    159+         DC    X'0' DATA CHECKS
000098 00                                    160+         DC    X'0' INTERVENTIONS
000099 00                                    161+         DC    X'0' NONTEXT TIMEOUTS
                                             162+*        THRESHOLDS
00009A FF                                    163+         DC    YL1(255) TRANSMISSIONS
00009B 0A                                    164+         DC    YL1(10) DATA CHECKS
00009C 05                                    165+         DC    YL1(5) INTERVENTIONS
00009D 05                                    166+         DC    YL1(5) NONTEXT TIMEOUTS
                                             167+*        RESERVED
00009E 0000                                  168+         DC    XL2'0'

                                             170          PRINT NOGEN

0000A0                                       172 BEGIN    EQU   *

                                             174 RTC      READ   MYDECB,TC,MYDCB,,,ANSRLIST,1,MF=E

0000CA 4590 C29A                002A0        187          BAL   9,TIO                     CHECK SIO CONDITION CODE
0000CE 47F0 C09A                000A0        188          B     RTC                       REISSUE READ CONNECT


0000D2                                       190 RTT      EQU   *
0000D2 5850 7014                00014        191          L     AREG,DECENTRY             POINT TO TERMINAL LIST
0000D6 D502 50C1 C812 000C1 00818            192          CLC   1(3,AREG),=X'000000'      WAS VALID ID RECEIVED
0000DC 4780 C10C                00112        193          BE    WTTD                      IF NOT, DISCONNECT THE LINE

                                             195          READ  MYDECB,TT,,,,,1,MF=E

0000FA 4590 C29A                002A0        204          BAL   9,TIO                     CHECK SIO CONDITION CODE
0000FE 47F0 C0CC                000D2        205          B     RTT                       REISSUE READ CONTINUE
                                                                                                         PAGE    4

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT                                      F30SEP69  10/06/70

000102 5850 700C                0000C        207          L     AREG,DECAREA              GET ADDRESS OF MESSAGE AREA
000106 9537 5000          00000              208          CLI   0(AREG),X'37'             HAS EOT BEEN RECEIVED
00010A 4780 C10C                00112        209          BE    WTTD                      IF SO, DISCONNECT THE LINE
00010E 47F0 C0CC                000D2        210          B     RTT                       IF NOT, READ MORE TEXT


                                             212 WTTD     WRITE MYDECB,TD,MF=E

000124 4590 C29A                002A0        219          BAL   9,TIO                     CHECK SIO CONDITION CODE
000128 47F0 C10C                00112        220          B     WTTD                      REISSUE WRITE DISCONNECT


00012C 41B0 C486                0048C        222          LA    AREAREG,OUTMSG            POINT TO OUTPUT MESSAGE

                                             224 WTC      WRITE  MYDECB,TC,MYDCB,(AREAREG),58,DIALLIST,1,MF=E

000166 4590 C29A                002A0        240          BAL   9,TIO                     CHECK SIO CONDITION CODE
00016A 47F0 C12A                00130        241          B     WTC                       REISSUE WRITE CONNECT
```

```
00016E 5850 7014              00014    243          L     AREG,DECENTRY            POINT TO TERMINAL LIST
000172 D502 5008 C815 00008 0081B      244          CLC   8(3,AREG),=X'E6E62D'     WAS ID RECEIVED FROM 2770-RR,ENQ
000178 4770 C17C              00182    245          BNE   NOT2770                  BRANCH IF NOT
00017C D201 B001 C80E 00001 00814      246          MVC   1(2,AREAREG),DC1SP       IF SO, MOVE THE DEVICE CONTROL
                                       247 *                                         CHARACTER FOR THE PRINTER
                                       248 *                                           INTO THE OUTPUT MESSAGE

000182                                 250 NOT2770  EQU   *

                                       252 WTT      WRITE MYDECB,TT,,(AREAREG),27,,1,MF=E

0001AC 4590 C29A              002A0    265          BAL   9,TIC                    CHECK SIO CONDITION CODE
0001B0 47F0 C17C              00182    266          B     WTT                      REISSUE WRITE CONTINUE


0001B4 41BB 001B              0001B    268          LA    AREAREG,27(AREAREG)      POINT TO NEXT OUTPUT MESSAGE
0001B8 58A0 C80A              00810    269          L     CTREG,COUNTER            UPDATE COUNT OF
0001BC 41AA 0001              00001    270          LA    CTREG,1(CTREG)             MESSAGES SENT
0001C0 50A0 C80A              00810    271          ST    CTREG,COUNTER            SAVE THE NEW COUNT
0001C4 9506 C80D        00813          272          CLI   COUNTER+3,X'06'          HAVE SIX MESSAGES BEEN SENT
0001C8 4770 C17C              00182    273          BNE   WTT                      IF NOT, POINT TO NEXT MESSAGE
0001CC D703 C80A C80A 00810 00810      274          XC    COUNTER(4),COUNTER       RESET COUNTER TO ZERO

                                       276 WTD      WRITE  MYDECB,TD,MF=E

0001E4 4590 C29A              002A0    283          BAL   9,TIC                    CHECK SIO CONDITION CODE
0001E8 47F0 C1CC              001D2    284          B     WTD                      REISSUE WRITE DISCONNECT


0001EC                                 286 SNAP       EQU   *
```

PAGE 5

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT                                  F30SEP69  10/06/70

                                     288          SNAP    ID=2,MF=(E,SDUMP)    LOOK AT ANSWER LIST BEFORE CHGNTRY

                                     294          CHGNTRY  ANSRLIST,SWLST,1,DISC  CHANGE CONTROL BYTE VALUE TO 1

                                     309          CHGNTRY   ANSRLIST,SWLST,2,DISC CHANGE CONTROL BYTE VALUE TO 1

                                     324          SNAP    ID=3,MF=(E,SDUMP)     LOOK AT ANSWER LIST AFTER CHGNTRY

                                     330          WTO     'SWTEST HAS SUCCESSFULLY COMPLETED'

00028C                               340 CLOSE    EQU   *

                                     342          CLOSE    (MYDCB)              CLOSE THE LINE DCB

                                     349          CLOSE  (SNAPDCB)              CLOSE THE SNAP DCB

000296                               356 EXIT     EQU   *
000296 58D0 C7C6           007CC     357          L     SAVEREG,SAVE+4            RESTORE REGISTERS

                                     359          RETURN    (14,12)             RETURN CONTROL

0002A0 12FF                          363 TIC      LTR   15,15                   EXCP ISSUED
0002A2 4780 C2CE           002D4     364          BZ    WAIT                    ISSUE WAIT IF GOOD SIO

                                     366          WTO   'SIO WAS NOT GOOD'

                                     376          SNAP   ID=4,MF=(E,SDUMP)      LOOK AT DECFLAGS

0002DC 47F0 C27A           00280     382          B     CLOSE


0002D4                               384 WAIT     EQU   *

                                     386          WAITR  1,ECB=(DECBREG)        WAIT FOR COMPLETION OF OPERATION

0002DC 9101 70C5     000C5           392          TM    DECTYPE+1,X'01'         IS THIS A READ OPERATION
0002E4 478C C37A           00380     393          BZ    WRTRTN                  IF NOT, GO TO WRITE ROUTINE
0002E8                               394 TESTCCDS EQU   *
0002E8 917F 70C0     000C0           395          TM    DECSDECB,X'7F'          WAS ECB POSTED NORMALLY
0002EC 47E0 C332           00338     396          BNC   BLERTN                  IF NOT, CHECK ERROR
0002F0 9500 7018     00018           397          CLI   DECFLAGS,X'00'          ARE ALL FLAGS ZERO
0002F4 4789 0004           00004     398          BE    4(9)                    IF SO, CONTINUE NORMALLY

                                     400          SNAP  ID=5,MF=(E,SDUMP)

000306 9510 7018     00018           406          CLI   DECFLAGS,X'10'          WAS AN INVALID ID RECEIVED
00030A 4770 C10C           00112     407          BNE   WTTD                    IF NOT, DISCONNECT LINE

                                     409          WTO   'AN INVALID ID WAS RECEIVED'

000334 47F0 C10C           00112     419          B     WTTD
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F30SEP69  10/06/70

000338                              421 BUFRTN   EQU   *

                                    423          SNAP  ID=6,MF=(E,SDUMP)        LOOK AT DECFLAGS

000346 9503 70C5        00005       429          CLI   DECTYPE+1,X'03'          IS THIS A READ CONTINUE
00034A 4770 C456              0045C 430          BNE   FINISH                   IF NOT, PRINT ERROR MESSAGE
00034E 91C8 7010        0001C       431          TM    DECSENSO,X'08'           WAS ERROR A DATA CHECK
000352 47E0 C456              0045C 432          BNO   FINISH                   IF NOT, PRINT ERROR MESSAGE
000356 58AC C8CA              0081C 433          L     CTREG,COUNTER            UPDATE COUNT OF
00035A 41AA 00C1              00001 434          LA    CTREG,1(CTREG)           ERRORS
00035E 50AC C80A              00810 435          ST    CTREG,COUNTER            SAVE NEW COUNT
000362 95C2 C8CD        00813       436          CLI   COUNTER+3,X'02'          IS COUNT OF ERRORS TWO
000366 47B0 C456              0045C 437          BNL   FINISH                   IF SO, WRITE ERROR MESSAGE

                                    439          READ  MYDECB,TP,MF=E

00037C 47F0 C29A              002A0 446          B     TIC                      CHECK SIO CONDITION CODE


000380 917F 7000        0000C       448 WRTRTN   TM    DECSDECB,X'7F'           WAS ECB POSTED NORMALLY
000384 47E0 C42C              00432 449          BNO   WRERR                    IF NOT, CHECK ERROR
000388 9500 7018        00018       450          CLI   DECFLAGS,X'00'           ALL FLAGS ZERO
00038C 4789 C0C4              00004 451          BE    4(9)                     IF SO, CONTINUE NORMALLY
000390 951C 70C5        000C5       452          CLI   DECTYPE+1,X'1C'          IS OPERATION WRITE CONNECT
000394 4770 C1CC              001D2 453          BNE   WTD                      IF NOT, DISCONNECT THE LINE
000398 95C0 7C18        00018       454          CLI   DECFLAGS,X'C0'           WAS WACK RECEIVED
00039C 4780 C3B6              003BC 455          BE    RETURN                   IF SO, CHECK FOR SECOND TIME
0003A0 585C 7C14              00014 456          L     AREG,DECENTRY            POINT TO TERMINAL LIST
0003A4 4155 00C8              00008 457          LA    AREG,8(AREG)             POINT TO READ-IN-AREA
0003A8 D5C2 50C0 C818 000C0 0081E   458          CLC   0(3,AREG),=X'D9D93D'     WAS ID NAK RECEIVED FROM 2780
0003AE 4780 C3B6              003BC 459          BE    RETURN                   IF SO, CHECK FOR SECOND TIME
0003B2 D5C2 5000 C81B 000C0 00821   460          CLC   0(3,AREG),=X'E6E63D'     WAS ID NAK RECEIVED FROM 2770
0003B8 4770 C3D4              003DA 461          BNE   FINI                     IF NOT, CHECK FOR INVALID ID
0003BC                              462 RETURN   EQU   *
0003BC 58AC C80A              00810 463          L     CTREG,COUNTER            UPDATE COUNT OF
0003C0 41AA 00C1              00001 464          LA    CTREG,1(CTREG)             TIMES RECEIVED
0003C4 50AC C8CA              00810 465          ST    CTREG,COUNTER            SAVE NEW COUNT
0003C8 95C2 C80D        00813       466          CLI   COUNTER+3,X'02'          HAS THIS BEEN RECEIVED TWICE
0003CC 4770 C12A              00130 467          BNE   WTC                      IF NOT, REISSUE WRITE CONNECT
0003D0 D703 C8CA C80A 00810 00810   468          XC    COUNTER(4),COUNTER       RESET COUNTER TO ZERO
0003D6 47F0 C1CC              001D2 469          B     WTD                      DISCONNECT THE LINE


0003DA                              471 FINI     EQU   *

                                    473          SNAP  ID=7,MF=(E,SDUMP)        LOOK AT THE SENSE IN THE DECB

0003E8 9510 7C18        00018       479          CLI   DECFLAGS,X'10'           WAS AN INVALID ID RECEIVED
0003EC 4770 C1CC              001D2 480          BNE   WTD                      IF NOT, DISCONNECT THE LINE

                                    482          WTO   'INVALID ID RECEIVED-NOTHING TRANSMITTED TO TERMINAL'

00042E 47F0 C1E6              001EC 492          B     SNAP
```



```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F30SEP69  10/06/70

000432                              494 WRERR    EQU   *

                                    496          SNAP  ID=8,MF=(E,SDUMP)

000440 9501 7010        0001C       502          CLI   DECSENSO,X'01'           WAS ERROR TIME OUT
000444 4770 C456              0045C 503          BNE   FINISH                   IF NOT, PRINT ERROR MESSAGE
000448 58AC C8CA              0081C 504          L     CTREG,COUNTER            UPDATE COUNT OF
00044C 41AA 00C1              00001 505          LA    CTREG,1(CTREG)             ERRORS
000450 50AC C8CA              0081C 506          ST    CTREG,COUNTER            SAVE NEW COUNT
000454 9502 C8CD        00813       507          CLI   COUNTER+3,X'02'          IS ERROR COUNT 2
000458 4749 C000              00000 508          BL    0(9)                     IF NOT, REISSUE OPERATION

                                    510 FINISH   WTO   'ERROR CANNOT BE HANDLED BY PROGRAM'

000488 47F0 C27A              00280 520          B     CLOSE
```

```
                                              522          PRINT GEN
00048C 0227C1                                 523 OUTMSG   DC    X'0227C1'
00048F E3C8C9E240C9E240                       524          DC    C'THIS IS TEST MESSAGE 1'
0004A5 1526                                   525          DC    X'1526'
0004A7 02                                     526          DC    X'02'
0004A8 E3C8C9E240C9E240                       527          DC    C'THIS IS TEST MESSAGE 2  '
0004C0 1526                                   528          DC    X'1526'
0004C2 02                                     529          DC    X'02'
0004C3 E3C8C9E240C9E240                       530          DC    C'THIS IS TEST MESSAGE 3  '
0004DB 1526                                   531          DC    X'1526'
0004DD 02                                     532          DC    X'02'
0004DE E3C8C9E240C9E240                       533          DC    C'THIS IS TEST MESSAGE 4  '
0004F6 1526                                   534          DC    X'1526'
0004F8 02                                     535          DC    X'02'
0004F9 E3C8C9E240C9E240                       536          DC    C'THIS IS TEST MESSAGE 5  '
000511 1526                                   537          DC    X'1526'
000513 02                                     538          DC    X'02'
000514 E3C8C9E240C9E240                       539          DC    C'THIS IS TEST MESSAGE 6  '
00052C 1526                                   540          DC    X'1526'

                                              542 *********************************************************************
                                              543 *                                                                   *
                                              544 *  IF THE USER WISHES TO EXECUTE THIS PROGRAM, HE MUST CODE HIS     *
                                              545 *  DFTRMLST MACROS TO HANDLE THE ID'S OF HIS PARTICULAR TERMINALS.  *
                                              546 *                                                                   *
                                              547 *********************************************************************

                                              549 ANSRLIST  DFTRMLST   SWLST,AN,4,0,2,1070,(D9D92D),(E6E62D)
000530                                        550+ANSRLIST DS     0F PUT ON FULL WORD BOUNDARY  000L
000530 FF                                     551+         DC     XL1'FF' IDENTIFY AS SWLST          000L
000531 000000                                 552+         DC     XL3'0' PTR TO MATCHING SEQUENCE   000L
000534 02                                     553+         DC     HL1'2' NUMBER OF LIST ENTRIES     000L
000535 04                                     554+         DC     HL1'4' ENTRY LENGTH               000L
000536 03                                     555+         DC     HL1'3' READ-IN-AREA LENGTH        000L
                                                                                                              PAGE    8

  LOC  OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT                                          F30SEP69  10/06/70

000537 00                                     556+         DC     HL1'0' NO. OF DIAL DIGITS         000L
000538 000000                                 557+         DC     XL3'0' READ-IN-AREA               000L
00053B 02                                     558+         DC     HL1'2' IDCOUNT                    000L
00053C 1070                                   559+         DC     XL2'1070' IDSENT                     000L
00053E D9D92D                                 560+         DC     X'D9D92D' AUTHORIZED SEQUENCE        000L
000541 00                                     561+         DC     XL1'00' CONTROL BYTE VALUE OF 0    000L
000542 E6E62D                                 562+         DC     X'E6E62D' AUTHORIZED SEQUENCE        000L
000545 00                                     563+         DC     XL1'00' CONTROL BYTE VALUE OF 0    000L

                                              565 DIALLIST   DFTRMLST   SWLST,MD,5,0,1,2D,(D9D91070,1),(E6E61070,1)
000548                                        566+DIALLIST DS     0F PUT ON FULL WORD BOUNDARY  000L
000548 FF                                     567+         DC     XL1'FF' IDENTIFY AS SWLST          000L
000549 000000                                 568+         DC     XL3'0' PTR TO MATCHING SEQUENCE   000L
00054C 02                                     569+         DC     HL1'2' NUMBER OF LIST ENTRIES     000L
00054D 05                                     570+         DC     HL1'5' ENTRY LENGTH               000L
00054E 04                                     571+         DC     HL1'4' READ-IN-AREA LENGTH        000L
00054F 00                                     572+         DC     HL1'0' NO. OF DIAL DIGITS         000L
000550 00000000                               573+         DC     XL4'0' READ-IN-AREA               000L
000554 01                                     574+         DC     HL1'1' IDCOUNT                    000L
000555 2D                                     575+         DC     XL1'2D' IDSENT                     000L
000556 D9D91070                               576+         DC     X'D9D91070' AUTHORIZED SEQUENCE         000L
00055A 01                                     577+         DC     YL1(1) CONTROL BYTE               000L
00055B E6E61070                               578+         DC     X'E6E61070' AUTHORIZED SEQUENCE         000L
00055F 01                                     579+         DC     YL1(1) CONTROL BYTE               000L

000560 0000000000000000                       581 INAREA   DC     100F'0'

                                              583 MYDCB    DCB    DSORG=CX,DEVD=BS,MACRF=(R,W),DDNAME=B2770DC1,                X
                                                                  LERB=ERRBLOCK,EROPT=TC

                                              585+*                          DATA CONTROL BLOCK
                                              586+*
0006DC                                        587+         ORG    *-20 TO ELIMINATE UNUSED SPACE
0006DC                                        588+MYDCB    DS     0F ORIGIN ON WORD BOUNDRY
0006F0                                        589+         ORG    *+20 TO ORIGIN GENERATION

                                              591+*                          COMMON ACCESS METHOD INTERFACE

0006F0 00                                     593+         DC     AL1(0) BUFNO
0006F1 000001                                 594+         DC     AL3(1) BUFCB
0006F4 0000                                   595+         DC     AL2(0) BUFL
0006F6 1000                                   596+         DC     BL2'0001000000000000' DSORG
0006F8 00000001                               597+         DC     A(1) IOBAD

                                              599+*                          FOUNDATION EXTENSION

0006FC 00                                     601+         DC     BL1'00000000' BFTEK,BFALN,HIARCHY
0006FD 18                                     602+         DC     BL1'00011000' BTAM EROPT CODE
0006FE FF                                     603+         DC     AL1(255) BTAM BUFFER COUNT
0006FF 00                                     604+         DC     AL1(0)
000700 00                                     605+         DC     BL1'00000000' RECFM
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F30SEP69  10/06/70

000701 000000                      606+          DC    AL3(0) EXLST

                                   608+*                       FOUNDATION BLOCK

000704 C2F2F7F7F0C4F0F1            610+          DC    CL8'B2770D01' DDNAME
00070C 02                          611+          DC    BL1'00000010' OFLGS
00070D 00                          612+          DC    BL1'00000000' IFLG
00070E 2020                        613+          DC    BL2'0010000000100000' MACR

                                   615+*                       BTAM INTERFACE

000710 0000008C                    617+          DC    A(ERRBLOCK) DCBLERB
000714 00                          618+          DC    BL1'00000000' MODE
000715 80                          619+          DC    BL1'10000000' MAS,CODE
000716 0000000000000000            620+          DC    XL26'0' CONTROL CHARS
000730 0000000000000000            621+          DC    4F'0' RESERVED


                                   624           READ    MYDECB,TT,MYDCB,INAREA,400,ANSRLIST,1,MF=L
000740                             625+          DS    0F
000740 00000000                    626+MYDECB    DC    A(0) EVENT CONTROL BLOCK
000744 00                          627+          DC    BL1'000'
000745 03                          628+          DC    AL1(3) TYPE FIELD
000746 0190                        629+          DC    AL2(400) LENGTH              OCCB
000748 000006DC                    630+          DC    A(MYDCB) DCB ADDRESS
00074C 0000056C                    631+          DC    A(INAREA) AREA ADDRESS       COOB
000750 00000000                    632+          DC    A(0) ERROR INFO. FIELD ADDR
000754 00000530                    633+          DC    A(ANSRLIST) TERMINAL LIST ADDRESS
000758 0001                        634+          DC    AL2(1) LINE NUMBER
00075A 0000                        635+          DC    AL2(0) RESPONSE FIELD
00075C 00                          636+          DC    AL1(0) TP-OP CODE
00075D 00                          637+          DC    AL1(0) ERROR STATUS
00075E 0000                        638+          DC    AL2(0) CSW STATUS
000760 00000000                    639+          DC    AL4(0) CURRENT ADDR LIST PTR
000764 00000000                    640+          DC    AL4(0) CURRENT ADDR POLL PTR


                                   642 SNAPDCB  DCB  DSORG=PS,RECFM=VBA,MACRF=W,BLKSIZE=1632,LRECL=125,        X
                                                     DDNAME=SNAPSW,DEVD=PR


                                   644+*                       DATA CONTROL BLOCK
                                   645+*
000758                             646+          ORG   *-16 TO ELIMINATE UNUSED SPACE
000758                             647+SNAPDCB   DS    0F ORIGIN ON WORD BOUNDRY
000768                             648+          ORG   *+16 TO ORIGIN GENERATION

                                   650+*                       PRINTER DEVICE INTERFACE

000768 0000                        652+          DC    BL2'0000000000000000' PRTSP,DEVT
00076A 0000                        653+          DC    H'0'

                                   655+*                       COMMON ACCESS METHOD INTERFACE
```



```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F30SEP69  10/06/70

00076C 00                          657+          DC    AL1(0) BUFNO
00076D 000001                      658+          DC    AL3(1) BUFCB
000770 0000                        659+          DC    AL2(0) BUFL
000772 4000                        660+          DC    BL2'0100000000000000' DSORG
000774 00000001                    661+          DC    A(1) IOBAD

                                   663+*                       FOUNDATION EXTENSION

000778 00                          665+          DC    BL1'00000000' BFTEK,BFALN,HIARCHY
000779 000001                      666+          DC    AL3(1) EODAD
00077C 54                          667+          DC    BL1'01010100' RECFM
00077D 000000                      668+          DC    AL3(0) EXLST

                                   670+*                       FOUNDATION BLOCK

000780 E2D5C1D7E2E64040            672+          DC    CL8'SNAPSW' DDNAME
000788 02                          673+          DC    BL1'00000010' OFLGS
000789 00                          674+          DC    BL1'00000000' IFLG
00078A 0020                        675+          DC    BL2'0000000000100000' MACR

                                   677+*                       BSAM-BPAM-QSAM INTERFACE

00078C 00                          679+          DC    BL1'00000000' RER1
00078D 000001                      680+          DC    AL3(1) CHECK, GERR, PERR
000790 00000001                    681+          DC    A(1) SYNAD
000794 0000                        682+          DC    H'0' CIND1, CIND2
000796 0660                        683+          DC    AL2(1632) BLKSIZE
000798 00000000                    684+          DC    F'0' WCPO, WCPL, OFFSR, OFFSW
00079C 00000001                    685+          DC    A(1) IOBA
0007A0 00                          686+          DC    AL1(0) NCP
0007A1 000001                      687+          DC    AL3(1) EOBR, EOBAD
```

```
                                       689+*                          BSAM-BPAM INTERFACE

0007A4 00000001                        691+         DC     A(1) EOBW
0007A8 0000                            692+         DC     H'0' DIRCT
0007AA 007D                            693+         DC     AL2(125) LRECL
0007AC 00000001                        694+         DC     A(1) CNTRL, NOTE, POINT

                                       696 SDUMP   SNAP   DCB=SNAPDCB,ID=1,PDATA=(REGS),
                                                          STORAGE=(ANSRLIST,SDUMP),MF=L
0007B0                                 697+SDUMP    DS     0F
0007B0 01                              698+         DC     AL1(1) ID NUMBER
0007B1 00                              699+         DC     AL1(0)
0007B2 86                              700+         DC     AL1(134) OPTION FLAGS
0007B3 20                              701+         DC     AL1(32) OPTION FLAGS
0007B4 00000758                        702+         DC     A(SNAPDCB) DCB ADDRESS
0007B8 00000000                        703+         DC     A(0) TCB ADDRESS
0007BC 000007C0                        704+         DC     A(*+4) ADDRESS OF SNAP-SHOT LIST
0007C0 00000530                        705+         DC     A(ANSRLIST) STARTING/ENDING ADDRESS
0007C4 000007B0                        706+         DC     A(SDUMP) STARTING/ENDING ADDRESS
0007C4                                 707+         ORG    *-4
0007C4 80                              708+         DC     X'80'
0007C8                                 709+         ORG    *+3
```



```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                       F30SEP69  10/06/70

0007C8                                 711 SAVE     DS     18F'0'                  SAVE AREA
00081C 00000000                        712 COUNTER  DC     F'0'                    COUNTER
000814 1140                            713 DC1SP    DC     X'1140'                 DC1,SPACE
000005                                 714 AREG     EQU    5                       REG USED TO POINT TO CFTRMLST
000006                                 715 DCBREG   EQU    6                       DCB REGISTER
000007                                 716 DECBREG  EQU    7                       DECB REGISTER
00000A                                 717 CTREG    EQU    10                      COUNTER REGISTER
00000B                                 718 AREAREG  EQU    11                      MESSAGE AREA REGISTER
00000C                                 719 BASEREG  EQU    12                      BASE REGISTER
00000D                                 720 SAVEREG  EQU    13                      SAVE AREA REGISTER

                                       722          DCBD   DEVD=BS,DSORG=BX

                                       724+*                          DCB SYMBOLIC DEFINITION FOR
                                       725+*                          BTAM LINE GROUP

000000                                 727+IHADCB   DSECT

                                       729+*                          BTAM LINE GROUP INTERFACE

000014                                 731+         ORG    IHADCB+20
000014                                 732+DCBBUFNC DS     0AL1
000014                                 733+DCBBUFCB DS     A
000018                                 734+DCBBUFL  DS     AL2
00001A                                 735+DCBDSORG DS     BL2
00001C                                 736+DCBIOBAD DS     A
000020                                 737+DCBHIARC DS     0BL1
000020                                 738+DCBBFTEK DS     BL1
000021                                 739+DCBERROP DS     AL1
000022                                 740+DCBBUFCT DS     AL1
000023                                 741+         DS     AL1
000024                                 742+DCBEXLST DS     A
00001C                                 743+         ORG    IHADCB+28
00001C                                 744+DCBDEVTP DS     BL1
000024                                 745+         ORG    IHADCB+36
000024                                 746+DCBEIOBX DS     BL1

                                       749+*                          FOUNDATION BEFORE OPEN

000028                                 751+         ORG    IHADCB+40
000028                                 752+DCBDDNAM DS     CL8
000030                                 753+DCBOFLGS DS     BL1
000031                                 754+DCBIFLG  DS     BL1
000032                                 755+DCBMACR  DS     BL2

                                       757+*                          FOUNDATION AFTER OPEN

000028                                 759+         ORG    IHADCB+40
000028                                 760+DCBTIOT  DS     BL2
00002A                                 761+DCBMACRF DS     BL2
00002C                                 762+DCBIFLGS DS     0BL1
00002C                                 763+DCBDEBAD DS     A
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F30SEP69  10/06/70

00003C                             764+DCBREAD  DS    0A
00003C                             765+DCBWRITE DS    0A

                                   767+*                      BTAM INTERFACE
000034                             768+         ORG   IHADCB+52
000034                             769+DCBLERB  DS    A

000038                             771+DCBXMCDF DS    BL1
000039                             772+DCBXCCDF DS    BL1
00003A                             773+DCBBSRSV DS    CL1
00003B                             774+DCBBSWPT DS    CL1
00003C                             775+DCBBSTSX DS    CL1
00003D                             776+DCBBSSTX DS    CL1
00003E                             777+DCBBSTEX DS    CL1
00003F                             778+DCBBSETX DS    CL1
000040                             779+DCBBSAK0 DS    CL2
000042                             780+DCBBSAK1 DS    CL2
000044                             781+DCBBSENQ DS    CL1
000045                             782+DCBBSNAK DS    CL1
000046                             783+DCBBSETB DS    CL1
000047                             784+DCBBSDLE DS    CL1
000048                             785+DCBBSEOT DS    CL1
000049                             786+DCBBSSYN DS    CL3
00004C                             787+DCBBSONL DS    CL2
00004E                             788+DCBBSSAK DS    CL2
000050                             789+DCBBSRVI DS    CL2

                                   791          IECTDECB
000000                             792+IECTDECB DSECT DECB DUMMY SECTION
                                   793+*        -----------------------------------------
                                   794+*        +                                       +
                                   795+*     0  +       STANDARD EVENT CONTROL BLOCK    +
                                   796+*        +                                       +
                                   797+*        +---------------------------------------+
                                   798+*        +                  +                    +
                                   799+*     4  +  OPERATION TYPE  +     AREA LENGTH    +
                                   800+*        +                  +                    +
                                   801+*        +---------------------------------------+
                                   802+*        + ON-LINE +                             +
                                   803+*     8  +TERMINAL +          ADDRESS OF DCB     +
                                   804+*        +  TEST   +                             +
                                   805+*        +---------------------------------------+
                                   806+*        +         +                             +
                                   807+*    12  +RESERVED +          ADDRESS OF AREA    +
                                   808+*        +         +                             +
                                   809+*        +---------------------------------------+
                                   810+*        +                  +                    +
                                   811+*    16  +    SENSE BYTES   +    RESIDUAL COUNT  +
                                   812+*        +     1   &   2    +                    +
                                   813+*        +---------------------------------------+
                                   814+*        +         +                             +
                                   815+*    20  + COMMAND +   ADDRESS OF TERMINAL LIST  +
                                   816+*        +   CODE  +                             +
                                   817+*        +---------------------------------------+
```



```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F30SEP69  10/06/70

                                   818+*        +         + RELATIVE+         +         +
                                   819+*    24  +  STATUS +   LINE  + ADDRESS + VRC/LRC +
                                   820+*        +  FLAGS  +  NUMBER + RESPONSE+ RESPONSE +
                                   821+*        +---------------------------------------+
                                   822+*        +         +         +                   +
                                   823+*    28  +  TP-OP  +  ERROR  +     CSW STATUS    +
                                   824+*        +   CODE  + STATUS  +                   +
                                   825+*        +---------------------------------------+
                                   826+*        +         +                             +
                                   827+*    32  +RESERVED +       ADDRESS OF CURRENT    +
                                   828+*        +         +        ADDRESSING ENTRY     +
                                   829+*        +---------------------------------------+
                                   830+*        +         +                             +
                                   831+*    36  +RESERVED +       ADDRESS OF CURRENT    +
                                   832+*        +         +          POLLING ENTRY      +
                                   833+*        +---------------------------------------+
                                   834+*        +         +         +                   +
                                   835+*    40  +RESERVED +RESERVED +  WRITE AREA LENGTH +
                                   836+*        +         +         +                   +
                                   837+*        +---------------------------------------+
                                   838+*        +         +                             +
                                   839+*    44  +RESERVED +   ADDRESS OF WRITE AREA     +
                                   840+*        +         +                             +
                                   841+*        -----------------------------------------
```

```
 LCC  CBJECT CODE     ACCR1 ACDR2  STMT   SCLRCE STATEMENT                                              F3CSEP69  1C/C6/70

C00000                            843+DECSCECB DS     1F STATUS FLAG + ADDRESS CF THE TCB

CCCCC4                            845+DECTYPE  DS     1H CPERATICN TYPE

C0C0C6                            847+DECLNGTH DS     1H AREA LENGTH

CCC0C8                            849+DECCNLTT DS     0CL1 RESERVEC FCR ON-LINE TERMINAL TEST
C00008                            85C+DECCCBAC DS     1F ADDRESS OF DCB

CCC0CC                            852+DECAREA  DS     1F ACDRESS CF AREA

C0C01C                            854+DECSENSC DS     1C 1ST SENSE BYTE

000011                            856+DECSENS1 DS     1C 2ND SENSE BYTE

00001Z                            858+DECCCLNT DS     1H RESICLAL CCUNT

C0C014                            86C+DECCMCCD DS     0CL1 COMMAND CODE
C00014                            861+DECENTRY DS     1F ACDRESS OF TERMINAL LIST

000C18                            864+DECFLAGS DS     1C STATUS FLAGS

00CC19                            866+DECRLN   DS     1C RELATIVE LINE NUMBER

000C1A                            868+DECRESPN DS     1H RESPCNSE FIELDS

CCC01C                            87C+DECTPCCD DS     1C TP-OP CCDE

00001D                            872+DECERRST DS     1C ERROR STATUS

00001E                            874+DECCSWST DS     1H CSW STATUS

00002C                            876+DECACRPT DS     1F ACCRESS CF CURRENT ADDRESSING ENTRY

000024                            878+DECPCLPT DS     1F ADDRESS CF CURRENT POLLING ENTRY

000028                            88C+         DS     2C RESERVED

00002A                            882+DECWLNG  DS     1H WRITE AREA LENGTH

00002C                            884+DECWAREA DS     1F ACCRESS CF WRITE AREA



                                  887          END
CC0818 CCC000                     888                 =X'CC0000'
CC081B E6E62D                     889                 =X'E6E62D'
00C81E C9C93D                     89C                 =X'C9D93D'
0C0821 E6E63D                     891                 =X'E6E63D'
```

CRCSS-REFERENCE


```
SYMBCL    LEN   VALUE  DEFN     REFERENCES                                                                      1C/C6/70

ANSRLIST 000C4 000530 00550   0180  C296  C3C4  0311  0319  0633  07C5
AREAREG  CCCC1 000C0B 0C718   0222  0232  C246  0258  C268  0268
AREG     00001 000005 0C714   0191  0192  C2C7  02C8  0243  0244  0456  C457  0457  0458  046C
BASEREG  C0001 0C0CCC 00719   0C99  01C0
BEGIN    CCCC1 0000A0 00172   0134
BLFRTN   000C1 000338 0C421   0396
CLCSE    0CCC1 00028C CC34C   0382  C520
CCUNTER  COCC4 000810 0C712   0269  0271  C272  0274  C274  0433  0435  0436  0463  0465  0466  0468  0468  0504  0506
                              0507
CTREG    0CCC1 000CCA 00717   0269  C270  C270  0271  C433  0434  C434  0435  0463  C464  0464  0465  05C4  C5C5  0505
                              0506
DCBBFTEK 0CCC1 000C20 CC738
DCBBSAKC 000C2 000040 00779
DCBBSAK1 CCCC2 0C0C42 0C78C
DCBBSDLE 0CC01 000047 C0784
DCBBSENC CCCC1 000044 C0781
DCBBSECT C00C1 000C48 CC785
DCBBSETB 0CCC1 000C46 0C783
DCBBSETX 0C0C1 00003F 0C778
DCBBSNAK C0CC1 00C045 0C782
DCBBSCNL 0CC02 000C4C 00787
DCBBSRSV C000C1 00003A 0C773
DCBBSRVI 000C2 000050 0C789
DCBBSSAK CCCC2 000C4E CC788
DCBBSSTX 000C1 00003D 0C776
DCBBSSYN 0CCC3 000C49 0C786
```

```
DCBBSTEX 00001 00003E 00777
DCBBSTSX 00001 00003C 00775
DCBBSWBT 00001 00003B 00774
DCBBUFCB 00004 000014 00733
DCBBUFCT 00001 000022 00740
DCBBUFL  00002 000018 00734
DCBBUFNO 00001 000014 00732
DCBDDNAM 00008 000028 00752
DCBDEBAD 00004 00002C 00763
DCBDEVTP 00001 00001C 00744
DCBDSORG 00002 00001A 00735
DCBEIOBX 00001 000024 00746
DCBERROP 00001 000021 00739
DCBEXLST 00004 000024 00742
DCBHIARC 00001 000020 00737
DCBIFLG  00001 000031 00754
DCBIFLGS 00001 00002C 00762
DCBIOBAD 00004 00001C 00736
DCBLERB  00004 000034 00769
DCBMACR  00002 000032 00755
DCBMACRF 00002 00002A 00761
DCBOFLGS 00001 000030 00753    0133
DCBREAD  00004 000030 00764
DCBREG   00001 000006 00715    0101  0106
DCBTIOT  00002 000028 00760
DCBWRITE 00004 000030 00765
DCBXCODE 00001 000039 00772
DCBXMODE 00001 000038 00771
DC1SP    00002 000814 00713    0246
```



```
SYMBOL    LEN   VALUE  DEFN     REFERENCES                                        10/06/70

DECADRPT 00004 000020 00876
DECAREA  00004 00000C 00852    0207
DECBREG  00001 000007 00716    0102  0107  0387
DECCMCOD 00001 000014 00860
DECCOUNT 00002 000012 00858
DECCSWST 00002 00001F 00874
DECDCBAD 00004 000008 00850
DECENTRY 00004 000014 00861    0191  0243  0456
DECERRST 00001 00001D 00872
DECFLAGS 00001 000018 00864    0397  0406  0450  0454  0479
DECLNGTH 00002 000006 00847
DECONLTT 00001 000008 00849
DECPOLPT 00004 000024 00878
DECRESPN 00002 00001A 00868
DECRLN   00001 000019 00866
DECSDECB 00004 00000C 00843    0395  0448
DECSENS0 00001 000010 00854    0431  0502
DECSENS1 00001 000011 00856
DECTPCOD 00001 00001C 00870
DECTYPE  00002 000004 00845    0392  0429  0452
DECWAREA 00004 00002C 00884
DECWLNG  00002 00002A 00882
DIALLIST 00004 000548 00566    0233
ENTRY    00002 000004 00099
ERRBLOCK 00004 00008C 00150    0617
EXAMPLE2 00001 000000 00001
EXIT     00001 000296 00356    0146
FINI     00001 0003DA 00471    0461
FINISH   00004 00045C 00512    0430  0432  0437  0503
IECTDECB 00001 000000 00792    0102
IHADCB   00001 000000 00727    0101  0731  0743  0745  0751  0759  0768
IHB0002  00001 00003A 00115    0112
IHB0002A 00002 00003A 00116    0111
IHB0005  00001 000086 00142    0139
IHB0005A 00002 000086 00143    0138
IHB0023  00001 00027D 00336    0333
IHB0023A 00002 00027E 00337    0332
IHB0027  00001 0002C0 00372    0369
IHB0027A 00002 0002C0 00373    0368
IHB0031  00001 000332 00415    0412
IHB0031A 00002 000332 00416    0411
IHB0036  00001 00042B 00488    0485
IHB0036A 00002 00042C 00489    0484
IHB0038  00001 000486 00516    0513
IHB0038A 00002 000486 00517    0512
INAREA   00004 00056C 00581    0631
MYDCB    00004 0006DC 00588    0106  0130  0178  0184  0230  0237  0346  0630
MYDECB   00004 00074C 00626    0107  0175  0196  0213  0225  0253  0277  0440
NOT2770  00001 000182 00250    0245
OUTMSG   00003 00048C 00523    0222
RETURN   00001 0003BC 00462    0455  0459
RTC      00004 0000AC 00175    0188
RTT      00001 0000D2 00190    0205  0210
SAVE     00004 0007C8 00711    0104  0105  0357
SAVEREG  00001 00000D 00720    0104  0105  0357
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F30SEP69  10/06/70

000000                                1 EXAMPLE1 CSECT
                                      2 ********************************************************************
                                      3 *                                                                  *
                                      4 *         SAMPLE IBM 2740/2760 PROGRAM                             *
                                      5 *                                                                  *
                                      6 *         THIS PROGRAM EXERCISES THE BTAM ONLINE TEST LOGIC FOR A  *
                                      7 *         NONSWITCHED POINT-TO-POINT LINE TO WHICH IS CONNECTED AN *
                                      8 *         IBM 2740 EQUIPPED WITH AN IBM 2760 OPTICAL IMAGE UNIT.   *
                                      9 *                                                                  *
                                     10 *         OPERATION - WHEN THE PROGRAM HAS BEEN STARTED, TWO MESSAGES *
                                     11 *         WILL BE TYPED ON THE SYSTEM CONSOLE:                     *
                                     12 *                                                                  *
                                     13 *               IXTBT022 IS LOADED AT XXXXXXXX                     *
                                     14 *               IXTBT022 FOR NONSWITCHED 2760 HAS STARTED.         *
                                     15 *                                                                  *
                                     16 *         THE TERMINAL OPERATOR CAN NOW DO ONE OF THE FOLLOWING:   *
                                     17 *                                                                  *
                                     18 *         1.) PROBE THE 'LOAD' UTILITY RESPONSE POINT OF THE 2760. *
                                     19 *               'LOAD' SHOULD NOT BE PROBED WHEN THE FILMSTRIP IS  *
                                     20 *               ALREADY LOADED.                                    *
                                     21 *                                                                  *
                                     22 *         2.) ENTER AN ON-LINE TERMINAL TEST REQUEST MESSAGE FOR ANY *
                                     23 *               2740- OR 2760-TYPE TEST.                           *
                                     24 *                                                                  *
                                     25 *         3.) PROBE THE 'UNLOAD' UTILITY RESPONSE POINT. THIS WILL *
                                     26 *               CAUSE THE FILM TO BE RETRACTED AND THE CARTRIDGE   *
                                     27 *               TO BE EJECTED.  YOU MAY NOW BEGIN AGAIN.           *
                                     28 *                                                                  *
                                     29 *         4.) ENTER A MESSAGE FROM THE 2740 KEYBOARD REQUESTING A  *
                                     30 *               CLOSEDOWN.  THE 6 CHARACTERS OF THE MESSAGE ARE    *
                                     31 *                     CLOSE#  WHERE #=EOT CHARACTER (UPPER CASE)   *
                                     32 *                                                                  *
                                     33 *         THE PROGRAM IGNORES ANY OTHER RESPONSE. THE FOLLOWING    *
                                     34 *         MESSAGE WILL BE PRINTED ON THE 2740 KEYBOARD.            *
                                     35 *              'THE IMMEDIATELY PRECEDING MESSAGE WAS NOT          *
                                     36 *               OF THE PROPER FORMAT.                              *
                                     37 *               YOU ARE EXPECTED TO PROBE UNLOAD OR LOAD,          *
                                     38 *               OR ENTER AN ONLINE TEST REQUEST MESSAGE.'          *
                                     39 *                                                                  *
                                     40 ********************************************************************
                                     41          PRINT NOGEN

000008                               43 WORKREG  EQU   8
000009                               44 DCBREG   EQU   9
00000C                               45 BASEREG  EQU   12
00000D                               46 SAVEREG  EQU   13                   POINTER TO SAVE AREA
00000E                               47 RETREG   EQU   14                   BAL RETURN REGISTER
00000F                               48 BRREG    EQU   15                   BAL BRANCH ADDRESS REGISTER
000000                               49 START    EQU   *

                                     51          SAVE  (14,12)              SAVE THE REGISTERS

000004 05C0                          55          BALR  BASEREG,0
000006                               56          USING *,BASEREG            ESTABLISH ADDRESSABILITY
000000                               57          USING IHADCB,DCBREG
```



```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F30SEP69  10/06/70

000006 50C0 C3F2            003F8    58          ST    BASEREG,BASE         COMPUTE THE
00000A F384 C3E6 C3F2 003EC 003F8    59          UNPK  UNPACK(9),BASE(5)      LOAD ADDRESS
000010 DC07 C3E6 C307 003EC 0030D    60          TR    UNPACK(8),TRTABLE-240    FOR THE
000016 D207 C034 C3E6 0003A 003EC    61          MVC   WTOP+30(8),UNPACK          WTO MESSAGE

                                     63 WTOP     WTO   'IXTBT022 IS LOADED AT

                                     73          WTO   'IXTBT022 FOR NONSWITCHED 2760 HAS STARTED'

00007C 50D0 C32E            00334    83          ST    SAVEREG,SAVEAREA+4
000080 41D0 C32A            00330    84          LA    SAVEREG,SAVEAREA
000084 4190 C5C2            005C8    85          LA    DCBREG,DCB           BASE FOR DCB
000088                               86 OPEN     EQU   *
000088 9200 C3E1      003E7          87          MVI   IOCODE,0             INDICATE OPEN INITIATED

                                     89          OPEN  DCB                  OPEN THE DCB

000096 9110 9030      00030          96          TM    DCBOFLGS,X'10'    DID OPEN COMPLETE SUCCESSFULLY
00009A 4770 C0D0            000D6    97          BNZ   OPENCK

                                     99          WTO   'OPEN DID NOT COMPLETE SUCCESSFULLY'
```

```
0000CC 5800 C32E            00334   109            SAVEREG,SAVEAREA+4

                                    111            RETURN   (14,12)           RESTORE THE REGISTERS

0000D6                              115 OPENCK     EQU    *
0000D6 9200 C3E0        003E6       116            MVI    ERRCRCT,0             ZERO ERROR COUNT

                                    118            TRNSLATE  DCB,SD40,ERRMSG,LENGTH   TRANSLATE ERROR MESSAGE

0000FE                              131 READTI     EQU    *
0000FE 9204 C3E1        003E7       132            MVI    ICCODE,4              INDICATE READ TI

                                    134            READ   READECB,TI,,INAREA,27,,1,MF=E

000130 12FF                         148            LTR    BRREG,BRREG           NORMAL INITIATION
000132 4770 C2C6             002CC  149            BNZ    ICERROR               BRANCH IF NO
000136 45E0 C1FA             00200  150            BAL    RETREG,WAIT
00013A D501 C372 C3DA 00378 003E0   151 COMPARE    CLC    INAREA(2),PREC        IS MESSAGE FROM 2760
000140 4770 C1AC             001B2  152            BNE    IGNORE                BRANCH IF NO
000144 D501 C376 C3DC 0037C 003E2   153            CLC    INAREA+4(2),UNLOAD    IS REQUEST TO UNLOAD
00014A 4780 C21A             00220  154            BE     OUT                   BRANCH IF YES
00014E D501 C376 C3DE 0037C 003E4   155            CLC    INAREA+4(2),LOAD      IS REQUEST TO LOAD
000154 4770 C1AC             001B2  156            BNE    IGNORE                IGNORE MESSAGE IF IT IS NOT
                                    157 *                                         LOAD OR UNLOAD
000158 95FF C3E2        003E8       158            CLI    LOADDONE,X'FF'        IS FILM ALREADY LOADED
00015C 4780 C1AC             001B2  159            BE     IGNORE                BRANCH IF YES
000160 D202 C372 C403 00378 00409   160            MVC    INAREA(3),LOADMSG     MOVE FAA FOR LOAD
000166                              161 WRITETCO   EQU *
000166 920C C3E1        003E7       162            MVI    ICCODE,12             INDICATE WRITE TCO

                                    164            WRITE  READECB,TCO,,INAREA,27,LOADMSG,1,MF=E

0001A0 12FF                         180            LTR    BRREG,BRREG           NORMAL INITIATION
```



```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                        F3CSEP69  10/06/70

0001A2 4770 C2C6             002CC  181            BNZ    ICERROR               BRANCH IF NO
0001A6 45E0 C1FA             00200  182            BAL    RETREG,WAIT
0001AA 92FF C3E2        003E8       183            MVI    LOADDONE,X'FF'        INDICATE FILM LOADED
0001AE 47F0 C134             0013A  184            B      COMPARE

0001B2                              186 IGNORE     EQU    *
0001B2 D504 C372 C407 00378 0040D   187            CLC    INAREA(5),CLOSE       IS REQUEST TO CLOSE LINE
0001B8 4780 C256             0025C  188            BE     END                   YES
0001BC 9210 C3E1        003E7       189            MVI    ICCODE,16             INDICATE WRITE TI

                                    191            WRITE  READECB,TIR,,ERRMSG,LENGTH,,1,MF=E

0001F2 12FF                         206            LTR    BRREG,BRREG           NORMAL INITIATION
0001F4 4770 C2C6             002CC  207            BNZ    ICERROR               BRANCH IF NO
0001F8 45E0 C1FA             00200  208            BAL    RETREG,WAIT
0001FC 47F0 C0F8             000FE  209            B      READTI

000200                              211 WAIT       EQU    *
000200 50E0 C3D6             003DC  212            ST     RETREG,TEMP

                                    214            WAIT   1,ECB=READECB

00020E 957F C5FA        00600       219            CLI    READECB,X'7F'         IS COMPLETION SATISFACTORY
000212 4770 C2C6        002CC       220            BNE    ICERROR               BRANCH IF NO
000216 9200 C3E0        003E6       221            MVI    ERRCRCT,0             RESET ERROR COUNT
00021A 58E0 C3D6             003DC  222            L      RETREG,TEMP           RESTORE RETURN REGISTER
00021E 07FE                         223            BR     RETREG                RETURN

000220                              225 OUT        EQU    *
000220 9208 C3E1        003E7       226            MVI    ICCODE,8              INDICATE WRITE TIO

                                    228            WRITE  READECB,TIO,,,3,REWIND,,MF=E

00024A 12FF                         240            LTR    BRREG,BRREG           NORMAL INITIATION
00024C 4770 C2C6             002CC  241            BNZ    ICERROR               BRANCH IF NO
000250 45E0 C1FA             00200  242            BAL    RETREG,WAIT
000254 9200 C3E2        003E8       243            MVI    LOADDONE,X'00'        INDICATE FILM NO LONGER LOADED
000258 47F0 C0F8             000FE  244            B      READTI

00025C                              246 END        EQU    *

                                    248            WTO    'IXTBT022 ENDED BY TERMINAL OPERATOR REQUEST'

                                    258            WRITE  READECB,TIO,,,3,REWIND,,MF=E

000288 12FF                         270            LTR    BRREG,BRREG           NORMAL INITIATION
00028A 4770 C2F8             002FE  271            BNZ    PERM                  BRANCH IF NO
```

```
  LCC  CBJECT CODE     ADDR1 ADDR2  STMT   SCURCE STATEMENT                                              F3CSEP69   10/06/70

C0028E 45E0 C1FA              00200    272              BAL    RETREG,WAIT
C002C2 58DC C32E              00334    273              L      SAVEREG,SAVEAREA+4        RESTCRE THE REGISTERS

                                       275              RETURN    (14,12)                RETURN

C002CC 9102 C3E0        003E6          279 ICERRCR  TM     ERRCRCT,2                 HAVE THERE BEEN TWC ERRORS
                                       280 *                                           CN THIS CCMMAND ALREADY
CC02DC 471C C2F8              002FE    281              BC     PERM                      YES - ABENC
C002D4 1B88                            282              SR     WCRKREG,WORKREG
C002D6 438C C3E0              003E6    283              IC     WCRKREG,ERRCRCT
C002DA 4188 0CC1              00001    284              LA     WCRKREG,1(WORKREG)        INCREMENT ERROR CCUNT BY 1
C002DE 4280 C3E0              003E6    285              STC    WCRKREG,ERRORCT
0002E2 4380 C3E1              003E7    286              IC     WCRKREG,ICCODE            PICK UP BRANCH INCEX
0002E6 47F8 C2E4              002EA    287              B      BRANCHT(WCRKREG)          RETURN TO RETRY
C002EA 47FC C082              00088    288 BRANCHT  B      CPEN
C002EE 47FC C0F8              000FE    289              B      READTI                    READ TI
C002F2 47FC C21A              C0220    290              B      CLT                       WRITE TIC
0002F6 47FC C160              00166    291              B      WRITETCC                  WRITE TCO
C002FA 47FC C1AC              001B2    292              B      IGNCRE                    WRITE TI


C002FE                                 294 PERM     ECU    *

                                       296              WTC    'IXTRTC22 ENDED - I/O ERRCR'

C00324 58D0 C32E              00334    306              L      SAVEREG,SAVEAREA+4        RESTCRE THE REGISTERS

                                       308              RETURN    (14,12)                RETURN



                                       312 ******** CCNSTANTS AREA
C0032E 0000
C0033C 0CCCCCCC00000000                313 SAVEAREA DC     18F'0'                    THIS ROUTINE'S SAVE AREA
C00378                                 314 INAREA   DS     25F
C003DC CCCCCCCC                        315 TEMP     CC     F'0'
0003EC 3E4C                            316 PREC     CC     X'3E4C'                   PREFIX 0
0003E2 45C1                            317 UNLCAD   DC     X'4501'                   UNLCAD COORDINATES
0003E4 25C1                            318 LCAC     CC     X'2501'                   LOAC COORDINATES
C003E6 00                              319 ERRCRCT  DC     X'CC'                     COUNT OF ABNORMAL COMPLETIONS
                                       320 *                                           CF LAST I/O REQUEST  MAX=2
C003E7 CC                              321 ICCCDE   DC     X'C0'                     C0 = CPEN INITIATED
                                       322 *                                         04 = READ INITIAL
                                       323 *                                         C8 = WRITE TIC
                                       324 *                                         0C = WRITE TCC
                                       325 *                                         1C = WRITE INITIAL
CC03E8 CC                              326 LCACCCNE DC     X'00'                     TURNED ON WHEN FILM HAS BEEN
                                       327 *                                           LCADED
C003E9 C000C0
CC03EC 0CCCCCCCCCC000000               328 UNPACK   DC     3F'0'
CC03F8 000C000CFF                      329 BASE     CC     XL5'000C0C0000FF'
CC03FD FOF1F2F3F4F5F6F7                330 TRTABLE  DC     XL16'F0F1F2F3F4F5F6F7F8F9C1C2C3C4C5C6'
```



```
  LCC  OBJECT CODE     ADDR1 ADDR2  STMT   SCURCE STATEMENT                                              F3CSEP69   10/06/70

C004CC E7C6CCA5EB                      331 CLCSE    DC     X'E7C6CCA5EB'
C00414                                 332 ERRMSG   DS     0F
C00414 1517171717171717                333              DC     X'15171717171717171717171717'  CR ANC ICLES
C0041F 40404CE3C8C54CC9                334              DC     C'   THE IMMEDIATELY PRECEDING MESSAGE WAS NCT '
C0044C 1517171717171717                335              DC     X'15171717171717171717171717'  CR AND ICLES
C00457 C6C640E3C8C540D7                336              DC     C'CF THE PROPER FORMAT.'
C0046C 1517171717171717                337              DC     X'15171717171717171717171717'  CR AND IDLES
C00477 E8D6E440C1D9C540                338              DC     C'YCU ARE EXPECTEC TO PROBE UNLCAD CR LOAD, '
C004A1 1517171717171717                339              DC     X'15171717171717171717171717'  CR AND ICLES
C004AC D6D940C5D5E3C5D9                340              DC     C'CR ENTER AN ONLINE TEST REQUEST MESSAGE. '
0004D5 26                              341              DC     X'26'                     EOB
C000C2                                 342 LENGTH   ECU    *-ERRMSG
0004D6 C16E6E                          343 REWIND   DC     X'C16E6E'                 MESSAGE TO UNLCAD FILM
0004D9 C20104                          344 LCADMSG  DC     X'C20104'                 MESSAGE TC LOAD FILM
                                       345              PRINT GEN
```

```
                                          347 ******** CONTROL BLOCKS

                                          349          ASMTRTAB  SD40
C004DC                                    350+IECTSD40 EQU    *
                                          351+*                  0 1 2 3 4 5 6 7 8 9 A B C D E F
0004DC 5E37163D887A7C7F                   352+         DC     X'5E37163D887A7C7F88888888885B8888' 0
0004EC 88888888885B5D5E                   353+         DC     X'88888888885B5D5E8888888888888888' 1
0004FC 88888888883B3D88                   354+         DC     X'88888888883B3D888888888888888888' 2
00050C 88885E8888881C1F                   355+         DC     X'88885E8888881C1F8888888888888888' 3
00051C C188888888888888                   356+         DC     X'C1888888888888888888A0768493E1B7' 4
00052C 6188888888888888                   357+         DC     X'61888888888888888888D757909587F6' 5
00053C 4023888888888888                   358+         DC     X'40238888888888888888888378BC08EA3' 6
00054C 8888888888888888                   359+         DC     X'888888888888888888888162C8D8296' 7
00055C 88626467686B6D6E                   360+         DC     X'88626467686B6D6E7C73888888888888' 8
00056C 88434546494A4C4F                   361+         DC     X'88434546494A4C4F5152888888888888' 9
00057C 8888252629 2A2C2F                  362+         DC     X'88882526292A2C2F3132888888888888' A
00058C 8888888888888888                   363+         DC     X'88888888888888888888888888888888' B
00059C 88E2E4E7E8EBEDEE                   364+         DC     X'88E2E4E7E8EBEDEEF0F3888888888888' C
0005AC 88C3C5C6C9CACCCF                   365+         DC     X'88C3C5C6C9CACCCFD1D2888888888888' D
0005BC 8888A5A6A9AAACAF                   366+         DC     X'8888A5A6A9AAACAFB1B2888888888888' E
0005CC 15C2C40708CB0D0E                   367+         DC     X'15C2C4C708CBCDCE1C13888888888888' F

                                          369 DCB      DCB    DSORG=CX,MACRF=(R,W),DDNAME=B2760L01,EROPT=WT

                                          371+*                      DATA CONTROL BLOCK
                                          372+*
0005C8                                    373+         ORG    *-20 TO ELIMINATE UNUSED SPACE
0005C8                                    374+DCB      DS     OF ORIGIN ON WORD BOUNDRY
0005DC                                    375+         ORG    *+20 TO ORIGIN GENERATION

                                          377+*                      COMMON ACCESS METHOD INTERFACE

0005DC 00                                 379+         DC     AL1(0) BUFNO
```



```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                            F30SEP69  10/06/70

0005DD 000001                             380+         DC     AL3(1) BUFCB
0005E0 0000                               381+         DC     AL2(0) BUFL
0005E2 1000                               382+         DC     BL2'0001000000000000' DSORG
0005E4 00000001                           383+         DC     A(1) IOBAD

                                          385+*                      FOUNDATION EXTENSION

0005E8 00                                 387+         DC     BL1'00000000' BFTEK,BFALN,HIARCHY
0005E9 14                                 388+         DC     BL1'00010100' BTAM EROPT CODE
0005EA FF                                 389+         DC     AL1(255) BTAM BUFFER COUNT
0005EB 00                                 390+         DC     AL1(0)
0005EC 00                                 391+         DC     BL1'00000000' RECFM
0005ED 000000                             392+         DC     AL3(0) EXLST

                                          394+*                      FOUNDATION BLOCK

0005F0 C2F2F7F6F0D3F0F1                   396+         DC     CL8'B2760L01' DDNAME
0005F8 02                                 397+         DC     BL1'00000010' OFLGS
0005F9 00                                 398+         DC     BL1'00000000' IFLG
0005FA 2020                               399+         DC     BL2'0010000000100000' MACR

                                          401+*                      BTAM INTERFACE

0005FC 00000000                           403+         DC     F'0'

000600                                    405          DS     0F

                                          407          READ   READECB,TI,DCB,,,,1,MF=L
000600                                    408+         DS     0F
000600 00000000                           409+READECB  DC     A(0) EVENT CONTROL BLOCK
000604 00                                 410+         DC     BL1'000'
000605 01                                 411+         DC     AL1(1) TYPE FIELD
000606 0000                               412+         DC     AL2(0) LENGTH
000608 000005C8                           413+         DC     A(DCB) DCB ADDRESS
00060C 00000000                           414+         DC     A(0) AREA ADDRESS
000610 00000000                           415+         DC     A(0) ERROR INFO. FIELD ADDR
000614 00000000                           416+         DC     A(0) TERMINAL LIST ADDRESS
000618 0001                               417+         DC     AL2(1) LINE NUMBER
00061A 0000                               418+         DC     AL2(0) RESPONSE FIELD
00061C 00                                 419+         DC     AL1(0) TP-OP CODE
00061D 00                                 420+         DC     AL1(0) ERROR STATUS
00061E 0000                               421+         DC     AL2(0) CSW STATUS
000620 00000000                           422+         DC     AL4(0) CURRENT ADDR LIST PTR
000624 00000000                           423+         DC     AL4(0) CURRENT ADDR POLL PTR

                                          425          IECTDECB
000000                                    426+IECTDECB DSECT  DECB DUMMY SECTION
                                          427+*        ----------------------------------------
                                          428+*        +                                      +
                                          429+*        +      STANDARD EVENT CONTROL BLOCK    +
                                          430+*        +                                      +
                                          431+*        +--------------------------------------+
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F30SEP69  10/06/70

                                 432+*        +                    +                   +
                                 433+*     4  +  OPERATION TYPE    +     AREA LENGTH   +
                                 434+*        +                    +                   +
                                 435+*        +----------------------------------------+
                                 436+*        + ON-LINE +                              +
                                 437+*     8  +TERMINAL +        ADDRESS OF DCB        +
                                 438+*        +  TEST   +                              +
                                 439+*        +----------------------------------------+
                                 440+*        +         +                              +
                                 441+*    12  +RESERVED +        ADDRESS OF AREA       +
                                 442+*        +         +                              +
                                 443+*        +----------------------------------------+
                                 444+*        +                    +                   +
                                 445+*    16  +     SENSE BYTES    +   RESIDUAL COUNT  +
                                 446+*        +     1    &    2    +                   +
                                 447+*        +----------------------------------------+
                                 448+*        +         +                              +
                                 449+*    20  + COMMAND |   ADDRESS OF TERMINAL LIST   +
                                 450+*        +   CODE  +                              +
                                 451+*        +----------------------------------------+
                                 452+*        +         + RELATIVE+          +         +
                                 453+*    24  +  STATUS +   LINE  + ADDRESS + VRC/LRC  +
                                 454+*        +  FLAGS  +  NUMBER + RESPONSE+ RESPONSE +
                                 455+*        +----------------------------------------+
                                 456+*        +         +         +                    +
                                 457+*    28  +  TP-OP  +  ERROR  +     CSW STATUS     +
                                 458+*        +   CODE  +  STATUS +                    +
                                 459+*        +----------------------------------------+
                                 460+*        +         +                              +
                                 461+*    32  +RESERVED +      ADDRESS OF CURRENT      +
                                 462+*        +         +       ADDRESSING ENTRY       +
                                 463+*        +----------------------------------------+
                                 464+*        +         +                              +
                                 465+*    36  +RESERVED +      ADDRESS OF CURRENT      +
                                 466+*        +         +         POLLING ENTRY        +
                                 467+*        +----------------------------------------+
                                 468+*        +         +         +                    +
                                 469+*    40  +RESERVED +RESERVED +  WRITE AREA LENGTH +
                                 470+*        +         +         +                    +
                                 471+*        +----------------------------------------+
                                 472+*        +         +                              +
                                 473+*    44  +RESERVED +    ADDRESS OF WRITE AREA     +
                                 474+*        +         +                              +
                                 475+*        ------------------------------------------
```



```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F30SEP69  10/06/70

000000                           477+DECSDECB DS    1F STATUS FLAG + ADDRESS OF THE TCB

000004                           479+DECTYPE  DS    1H OPERATION TYPE

000006                           481+DECLNGTH DS    1H AREA LENGTH

000008                           483+DECONLTT DS    0CL1 RESERVED FOR ON-LINE TERMINAL TEST
000008                           484+DECDCBAD DS    1F ADDRESS OF DCB

00000C                           486+DECAREA  DS    1F ADDRESS OF AREA

000010                           488+DECSENS0 DS    1C 1ST SENSE BYTE

000011                           490+DECSENS1 DS    1C 2ND SENSE BYTE

000012                           492+DECCOUNT DS    1H RESIDUAL COUNT

000014                           494+DECCMCOD DS    0CL1 COMMAND CODE
000014                           495+DECENTRY DS    1F ADDRESS OF TERMINAL LIST


000018                           498+DECFLAGS DS    1C STATUS FLAGS

000019                           500+DECRLN   DS    1C RELATIVE LINE NUMBER

00001A                           502+DECRESPN DS    1H RESPONSE FIELDS

00001C                           504+DECTPCOD DS    1C TP-OP CODE

00001D                           506+DECERRST DS    1C ERROR STATUS

00001E                           508+DECCSWST DS    1H CSW STATUS

000020                           510+DECADRPT DS    1F ADDRESS OF CURRENT ADDRESSING ENTRY

000024                           512+DECPOLPT DS    1F ADDRESS OF CURRENT POLLING ENTRY

000028                           514+         DS    2C RESERVED

00002A                           516+DECWLNG  DS    1H WRITE AREA LENGTH

00002C                           518+DECWAREA DS    1F ADDRESS OF WRITE AREA
```

```
                                   521          DCBD
                                   522               *,***  IHB068  NO VALID DSORG SPECIFIED-EXCP ASSUMED

                                   524+*                    DCB SYMBOLIC DEFINITION FOR
                                   525+*                    BASIC EXCP
000000                             527+IHADCB   DSECT
```



```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                F03SEP69  10/06/70
                                  531+*                       FOUNDATION BEFORE OPEN

000028                            533+         ORG    IHADCB+40
000028                            534+DCBDDNAM DS     CL8
000030                            535+DCBOFLGS DS     BL1
000031                            536+DCBIFLG  DS     BL1
000032                            537+DCBMACR  DS     BL2

                                  539+*                       FOUNDATION AFTER OPEN

000028                            541+         ORG    IHADCB+40
000028                            542+DCBTIOT  DS     BL2
00002A                            543+DCBMACRF DS     BL2
00002C                            544+DCBIFLGS DS     0BL1
00002C                            545+DCBDEBAD DS     A

000030                            548          CNOP   0,8

                                  550          END
```



```
SYMBOL    LEN   VALUE  DEFN    REFERENCES                                                    10/06/70

BASE      00005 0003F8 00329   0058  0059
BASEREG   00001 0000CC 00045   0055  0056  0058
BRANCHT   00004 0002EA 00288   0287
BRREG     00001 00000F 00048   0148  0148  0180  0180  0206  0206  0240  0240  0270  0270
CLOSE     00005 00040D 00331   0187
COMPARE   00006 00013A 00151   0184
DCB       00004 0005C8 00374   0085  0093  0123  0413
DCBDDNAM  00008 000028 00534
DCBDEBAD  00004 00002C 00545
DCBIFLG   00001 000031 00536
DCBIFLGS  00001 00002C 00544
DCBMACR   00002 000032 00537
DCBMACRF  00002 00002A 00543
DCBOFLGS  00001 000030 00535   0096
DCBREG    00001 000009 00044   0057  0095
DCBTIOT   00002 000028 00542
DECADRPT  00004 000020 00510
DECAREA   00004 00000C 00486
DECCMCOD  00001 000014 00494
DECCOUNT  00002 000012 00492
DECCSWST  00002 00001E 00508
DECDCBAD  00004 000008 00484
DECENTRY  00004 000014 00495
DECERRST  00001 00001D 00506
DECFLAGS  00001 000018 00498
DECLNGTH  00002 000006 00481
DECONLTT  00001 000008 00483
DECPOLPT  00004 000024 00512
DECRESPN  00002 00001A 00502
DECRLN    00001 000019 00500
DECSDECB  00004 000000 00477
DECSENS0  00001 000010 00488
DECSENS1  00001 000011 00490
DECTPCOD  00001 00001C 00504
DECTYPE   00002 000004 00479
DECWAREA  00004 00002C 00518
DECWLNG   00002 00002A 00516
END       00001 00025C 00246   0188
ERRMSG    00004 000414 00332   0125  0198  0342
ERRORCT   00001 0003E6 00319   0116  0221  0279  0283  0295
EXAMPLE1  00001 000000 00001
IECTDECB  00001 000000 00426
IECTSD40  00001 0004DC 00350   0124
IGNORE    00001 0001B2 00186   0152  0156  0159  0292
IHADCB    00001 000000 00527   0057  0533  0541
```

```
IHBOCC2  00CC1 000046 0CC69   CC66
IHBOOC2A 00CC2 000046 0C07C   0065
IHBOCC3  00CC1 000079 0CC79   0076
IHBOCC3A COCC2 00007A 00C8C   0075
IHBOCC5  0CCC1 0000CA 00105   0102
IHBCCC5A 0CC02 0000CA 00106   0101
IHBOC18  CCCC1 00028F 00254   0251
IHBOC18A COCC2 000290 00255   0250
IHBOC22  CCCC1 000322 0C3C2   0299
IHBOC22A 00002 000322 00303   0298
```

```
                                        CRCSS-REFERENCE                                                          PAGE    2

SYMBCL    LEN  VALUE  CFFN     REFERENCES                                                                        10/06/7C

INAREA   CC004 0C0378 00314   0140  C151  C153  C155  C16C  017C  0187
ICCODE   0CCC1 0003E7 00321   0C87  C132  C162  0189  0226  0286
ICERRCR  0CCC4 0002CC 00279   0149  C181  C2C7  0220  C241
LENGTH   0CCC1 000CC2 00342   0120  C196
LCAD     C0C02 0C03E4 00318   0155
LCADCCNE 00001 0003E8 00326   0158  0183  C243
LCADMSG  CCCC3 0004C9 0C344   016C  C172
CPEN     C0001 000088 00C86   0288
CPENCK   C00C1 000CD6 0C115   0097
CUT      CCCC1 00022C 0C225   0154  C29C
PERM     00C01 0002FE 0C294   0271  0281
PREO     000C2 0003EC 0C316   0151
READECB  CCCC4 00060C 004C9   0135  C165  C192  0215  0219  0229  0259
READTI   00001 0000FE 00131   0209  0244  C289
RETREG   0CCC1 0C0CCE 0CC47   C150  C182  C2C8  0212  0222  C223  0242  C272
REWINC   00C03 0004D6 00343   0234  0264
SAVEAREA 00CC4 000330 00313   0083  CC84  C1C9  0273  C3C6
SAVEREG  00001 00000D 00046   0083  0084  C1C9  0273  03C6
START    CCCC1 000CCC 0CC49
TEMP     00CC4 0003DC 00315   0212  0222
TRTABLE  00016 0003FD 0033C   0C6C
UNLOAD   00002 0003E2 00317   0153
UNPACK   CCCC4 C003EC 00328   0059  C06C  CC61
WAIT     000C1 000200 00211   015C  C182  C2C8  0242  0272
WCRKREG  CCCC1 000008 0CC43   0282  C282  C283  0284  C284  C285  C286  C287
WRITETCO 000C1 000166 00161   0291
WTCP     CCCC4 00001C 00065   CC61


NO STATEMENTS FLAGGED IN THIS ASSEMPLY
*STATISTICS*     SOURCE RECORDS (SYSIN) =    237      SCURCE RECCRDS (SYSLIB) =  4799
*CPTICNS IN EFFECT*   LIST, DECK, NOLCAD, NCRENT, XREF, NCTEST, ALGN, OS, LINECNT =  55
  469 PRINTEC LINES
```

The local 3270 sample program shows how BTAM support for the local 3270 display system works and demonstrates some of the capabilities of the display system. The sample program can operate from one to 255 local 3270 devices, at least one of which must be a 3277 display station with a keyboard. If more than one device is used, some or all of the remaining devices can be 3284 or 3286 printers (although the number of display stations should equal or exceed the number of printers). The sample program can work with model 1 or model 2 devices or both.

The local 3270 sample program is distributed as a member of SYS1.SAMPLIB named SAMP327L. This PDS member contains (in the form of 80-character card images) all the source statements for the sample program and most of the JCL needed to assemble, link-edit, and execute the program.

DEFINING THE LOCAL 3270 DISPLAY SYSTEM

The local 3270 display system used by the sample program is defined in two ways. The data definition (DD) statement DD3270 (and any other DD statements concatenated with it) specifies the devices to be used and assigns relative line numbers to them. The PARM parameter for the execution of the program describes the size and composition of the display system.

The DD statement DD3270 should be added to the JCL for the GO step (see the examples below). The UNIT parameter, which is the only required parameter, specifies the device to be used by device name (3277, 3284, or 3286), by device address, or by a UNITNAME defined during system generation. If more devices than one are to be used, DD statements for the additional devices should be concatenated with DD3270. Print output generated by a display station is directed to the printer specified by the DD statement immediately preceding the DD statement for the display station. (DD statements for printers should be separated by one or more DD statements for display stations.) If no printer is associated with a display station in this way, print output from that display station is directed to SYSPRINT.

Example 1: One 3270 device is used; it is attached to address 240:

```
    .
    .
    .
//GO.SYSABEND  DD SYSOUT=A
//GO.DD3270    DD UNIT=240
/*
```

<u>Example 2</u>: Three 3270 devices are used, two 3277 display stations and one 3286 printer; they are specified by device name:

```
    .
    .
    .
```

```
//GO.SYSABEND  DD SYSOUT=A
//GO.DD3270     DD UNIT=3277   relative line number 1
//              DD UNIT=3286   relative line number 2
//              DD UNIT=3277   relative line number 3
 /*
```

Print output from the 3277 display station on relative line number 1 is directed to SYSPRINT. Print output from the 3277 display station on relative line number 3 is directed to the printer on relative line number 2.

The PARM parameter of the EXEC statement is used to specify the number of devices to be used, the number of printers to be used, and the relative line numbers of the printers. The format of the EXEC statement with the PARM parameter is:

```
//    EXEC ASMFCLG,GO.PARM='number of devices[,number of
           printers[,printer rln]...]
```

Defaults and limits for these parameters are:

Number of devices: default = 1; minimum = 1; maximum = 255

Number of printers: default = 0; minimum = 0; maximum = one less than number of devices

Printer rln: if number of printers is zero = 0;
if number of printers is not zero, default = 1;
maximum = number of devices

Parameters are separated by commas. Any invalid characters are treated as zeros. If a parameter is longer than three characters, the rightmost three characters are used, and any others are ignored.

Example 3: A one-device system with a 3277 display station and no printer (as specified in example 1):

```
//    EXEC ASMFCLG
```

The EXEC statement supplied with the sample program can be used, since all the required values are defaults.

Example 4: A three-device system with two 3277 display stations and a printer on relative line number 2 (as specified in example 2):

```
//    EXEC ASMFCLG,GO.PARM='3,1,2'
```

Example 5: A four-device system with three 3277 display stations and a printer on relative line number 1:

```
//    EXEC ASMFCLG,GO.PARM='4,1'
```

The default for the relative line number of the printer is used.

Example: A two-device system with two 3277 display stations and no printer:

```
//    EXEC ASMFCLG,GO.PARM='2'
```

OPERATING THE SAMPLE PROGRAM

When the sample program is initialized, the initial format shown in Figure 43a or 43b is displayed on each display station. The subsequent operation of the program is controlled by the display station operator by means of the ENTER, CLEAR, PA1, and PA2 keys.

Pressing the CLEAR key (except when the ending format is displayed) causes the control options format shown in Figure 45a or 45b to be displayed.

Pressing the PA2 key (except when the ending format is displayed) causes the ending format shown in Figure 46a or 46b to be displayed. Further input from the display station is inhibited. After the PA2 key has been pressed on each display station, the sample program is terminated. (Any unstarted printer operations are lost when the program is terminated.)

The result of pressing the ENTER key or PA1 key depends on the format being displayed:

1. Initial format (Figure 43a or 43b): This is the first format displayed when the program is initialized. It can be redisplayed as described under the control options format. The CLEAR, PA2, and ENTER key are valid for this format.

   There are five unprotected fields on the screen; the field following "ZIP:" is numeric-only; Data may be entered into one or more of these fields. Then the ENTER key should be pressed to transmit the data from the display station to the program, which initializes an internal data area associated with the display station.

   The data is displayed as part of the verification format after the ENTER key has been pressed.

2. Verification format (Figure 44a or 44b): This format displays the data in the data area for the display station. The display station operator can verify, modify, or print the data. The CLEAR, PA2, ENTER, and PA1 keys are valid for this format.

   The ENTER key is used to transmit any modifications to the data to the program, which updates the data area for the display station.

   The PA1 key is used to have the data in the data area printed.

   The verification format is redisplayed after the ENTER or PA1 key has been pressed.

3. Control options format (Figure 45a or 45b): This format is displayed after the CLEAR key has been pressed. The CLEAR, PA2, ENTER, and PA1 keys are valid for this format.

   Pressing the ENTER key causes the verification format to be displayed.

   Pressing the PA1 key clears the data area for the display station and causes the initial format to be displayed.

   Also, the display station operator can enter a request for a BTAM online test (OLT) pattern by following the directions on the format. The pattern may be sent to any display station or printer being operated by the sample program. To continue, the CLEAR key should be pressed after an OLT pattern is sent or received to redisplay the control options format.

4. Ending format (Figure 46a or 46b): This format is displayed after the PA1 key has been pressed.

   Further input from the display station is inhibited.

If a key that is not valid is pressed, input from the display station is inhibited by the display station hardware and ignored by the sample program. To continue, the display station operator should press the RESET key to manually enable the keyboard; he should then press a valid key for the format being displayed.

```
ENTER DATA REQUESTED BELOW:

NAME:
ADDR:
CITY:
STATE:                                  ZIP:


ENTER KEY:  ENTER DATA;
PA2 KEY:  END PROGRAM;
CLEAR KEY:  CONTROL OPTIONS.
```

Figure 43A. Initial Format on Model 1 3277 Display Station

```
ENTER DATA REQUESTED BELOW:
NAME:                                   ADDR:                          ZIP:
CITY:                                   STATE:

ENTER KEY:  ENTER DATA;                 PA2 KEY:  END PROGRAM;
CLEAR KEY:  CONTROL OPTIONS.
```

Figure 43B. Initial Format on Model 2 3277 Display Station

```
DATA GIVEN BELOW ENTERED:

NAME:
ADDR:
CITY:
STATE:                                  ZIP:



ENTER KEY:  UPDATE DATA;
PA1 KEY:  PRINT DATA;
PA2 KEY:  END PROGRAM;
CLEAR KEY:  CONTROL OPTIONS.
```

Figure 44A. Verification Format on Model 1 3277 Display Station

```
DATA GIVEN BELOW ENTERED:
NAME:                                   ADDR:
CITY:                                   STATE:                      ZIP:


ENTER KEY:  UPDATE DATA;                PA1 KEY:  PRINT DATA;
PA2 KEY:  END PROGRAM;                  CLEAR KEY:  CONTROL OPTIONS.
```

Figure 44B. Verification Format on Model 2 3277 Display Station

```
XXYY3CUU

ENTER KEY:  RESUME AND CONTINUE;
PA1 KEY:  BEGIN NEW ENTRY;
PA2 KEY:  END PROGRAM;

TO REQUEST BTAM OLT -- ENTER REQUEST FOR TEST
  MESSAGE OVER SAMPLE FORMAT ABOVE:
  XX=TEST NO. (23-28)    YY=REPEATS  (01-99)
  CUU=ADDRESS OF THE TARGET DEVICE
  THEN HIT ERASE EOF AND THEN TEST REQ.
USE CLEAR KEY TO RESUME AFTER TEST.
```

Figure 45A. Control Options Format on Model 1 3277 Display Station

```
XXYY3CUU
ENTER KEY:  RESUME AND CONTINUE;        PA1 KEY:  BEGIN NEW ENTRY;
PA2 KEY:  END PROGRAM;
TO REQUEST BTAM OLT -- ENTER REQUEST FOR TEST MESSAGE OVER SAMPLE FORMAT ABOVE:
  XX=TEST NO.  (23-28)     YY=REPEATS  (01-99)   CUU=ADDRESS OF TARGET DEVICE
  THEN HIT ERASE EOF AND THEN TEST REQ.   USE CLEAR KEY TO RESUME AFTER TEST.
```

Figure 45B. Control Options Format on Model 2 3277 Display Station

```
_

        LOCAL 3270 SAMPLE PROGRAM ENDED.
```

Figure 46A. Ending Format on Model 1 3277 Display Station

```
                                        LOCAL 3270 SAMPLE PROGRAM ENDED.
```

Figure 46B. Ending Format on Model 2 3277 Display Station

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71    3/22/72

000000                                  2 SAMP327L CSECT
                                        3 * STATUS - CHANGE LEVEL 000
                                        4 *
                                        5 * ENTRY POINT - SAMP327L
                                        6 *
                                        7 * INPUT - REGISTERS 1, 13, AND 14 PROVIDE INPUT AS FOLLOWS
                                        8 *     1 - ADDRESS OF PARM AREA POINTER
                                        9 *     13- ADDRESS OF SAVE AREA
                                       10 *     14- RETURN ADDRESS
                                       11 *
                                       12 *     THE PARM AREA HAS THE FOLLOWING FORMAT
                                       13 *     +0  LENGTH FIELD (HALF WORD)
                                       14 *     +2  NO. OF DEVICES IN LINE GROUP (1-3 DIGITS,1-255, 1=DEFAULT)
                                       15 *         NO. OF PRINTERS IN LINE GROUP (1-3 DIGITS,<NO. DEVICES, 0=DEF)
                                       16 *         RLN'S OF PRINTERS (1-3 DIGITS,NO LARGER THAN NO. OF DEVICES,
                                       17 *              DEFAULT OF 1 IF NO. OF PRINTERS =1)
                                       18 *     ALL FIELDS SEPARATED BY COMMAS
                                       19 *
                                       20 *     DDNAME FOR 3270 DEVICES = DD3270
                                       21 *         DD CARDS ARRANGED - PRINTER, DISPLAYS, PRINTER, DISPLAYS, ETC.
                                       22 *         THOSE FOR PRINTERS OMITTED IF NO PRINTERS
                                       23 *
                                       24 *     SYSPRINT  DD  SYSOUT=A  REQUIRED
                                       25 *
                                       26 * OUTPUT - DISPLAY OUTPUT, PRINTER OUTPUT
                                       27 *
000000                                 28 ZEROREG  EQU   0
000001                                 29 PARMREG  EQU   1                   PARAMETER REGISTER
000002                                 30 EVENREG  EQU   2
000003                                 31 ODDREG   EQU   3
000004                                 32 LOOPREG  EQU   4
000005                                 33 INDXREG  EQU   5
000006                                 34 CNTREG   EQU   6
000006                                 35 RLNREG   EQU   CNTREG
000007                                 36 LOOPREG2 EQU   7
000007                                 37 DECBREG  EQU   7
000008                                 38 INDXREG2 EQU   8
000008                                 39 BUFREG   EQU   8
000009                                 40 TABBASE  EQU   9                   BASE FOR CONTROL TABLE
00000A                                 41 BASEREG2 EQU   10
00000B                                 42 BASEREG  EQU   11                  CSECT BASE REGISTER
00000D                                 43 SAVEREG  EQU   13
00000E                                 44 RETNREG  EQU   14
00000F                                 45 EPREG    EQU   15
                                       46 *
000005                                 47 TYPE     EQU   5                   DISP TO TYPE FIELD IN DECB
000012                                 48 COUNT    EQU   18                  DISP TO RESIDUAL COUNT IN DECB
000019                                 49 RLN      EQU   25                  DISP TO RLN FIELD IN DECB
000024                                 50 POLPT    EQU   36                  DISP TO POLPT FIELD IN DECB
                                       51 *
                                       52          SAVE  (14,12),T,*         SAVE REGISTER
000000 47F0 F00E            0000E      53+         B     14(0,15) BRANCH AROUND ID
000004 08                              54+         DC    AL1(8)
000005 E2C1D4D7F3F2F7D3                55+         DC    CL8'SAMP327L' IDENTIFIER
00000D 00
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71   3/22/72

00000E 90EC D00C            0000C   56+          STM   14,12,12(13) SAVE REGISTERS
000012 05B0                         57           BALR  BASEREG,0          ESTABLISH BASE REG
000014                              58           USING *,BASEREG,BASEREG2  & ADDRESSABILITY
000014 41AB 0FFF            00FFF   59           LA    BASEREG2,4095(BASEREG)   INITIALIZE SECOND BASE
000018 41AA 0001            00001   60           LA    BASEREG2,1(BASEREG2)
00001C 50D0 BA3C            00A50   61           ST    SAVEREG,SAVE+4     ESTABLISH
000020 4120 BA38            00A4C   62           LA    EVENREG,SAVE               SAVE AREA
000024 502D 0008            00008   63           ST    EVENREG,8(SAVEREG)                CHAIN
000028 18D2                         64           LR    SAVEREG,EVENREG    ESTABLISH NEW SAVE AREA
                                    65 *
00002A 5831 0000            00000   66           L     ODDREG,0(PARMREG)  GET ADDR OF PARM AREA
00002E 4843 0000            00000   67           LH    LOOPREG,0(ODDREG)  GET LENGTH OF PARM AREA
000032 1244                         68           LTR   LOOPREG,LOOPREG    PARMS PASSED
000034 4780 B054            00068   69           BZ    DEFAULTS           NO, TAKE DEFAULTS
000038 4153 0002            00002   70           LA    INDXREG,2(ODDREG)  INITIALIZE POINTER TO PARM STRING
00003C 45E0 B8FA            0090E   71           BAL   RETNREG,GETPARM    GET NUMBER OF DEVICES
000040 1211                         72           LTR   PARMREG,PARMREG    VALUE LESS THAN 1
000042 47D0 B03E            00052   73           BNP   DFLTNO             YES, TREAT AS 1
000046 5910 BA80            00A94   74           C     PARMREG,MAXRLN     VALUE GREATER THAN 255
00004A 4720 B03E            00052   75           BH    DFLTNO             YES, TREAT AS 1
00004E 5010 BA84            00A98   76           ST    PARMREG,NODVCS     SAVE NUMBER OF DEVICES
                                    77 *
000052 1244                         78 PARM2     LTR   LOOPREG,LOOPREG    ANY MORE PARMS
000052                              79 DFLTNO    EQU   PARM2              DEFAULT NO OF DVCS = 1
000054 4780 B054            00068   80           BZ    DFLTPTR            NO, DEFAULT NUMBER OF PRINTERS
                                    81 *
000058 45E0 B8FA            0090E   82           BAL   RETNREG,GETPARM    GET NUMBER OF PRINTERS
00005C 5910 BA84            00A98   83           C     PARMREG,NODVCS     NUMBER OF PRINTERS LESS THAN
000060 47B0 B054            00068   84           BNL   DFLTPTR            NUMBER OF DEVICES
000064 5010 BA88            00A9C   85           ST    PARMREG,NOPTRS     SAVE NUMBER OF PRINTERS
                                    86 *
000068 5820 BA84            00A98   87 GETTAB    L     EVENREG,NODVCS     GET SIZE OF TERMINAL ENTRIES
                                    88 *
000068                              89 DEFAULTS  EQU   GETTAB
000068                              90 DFLTPTR   EQU   GETTAB             DEFAULT NO OF PTRS = 0
                                    91 *
00006C 8B20 0003            00003   92           SLA   EVENREG,3
000070 5830 BA88            00A9C   93           L     ODDREG,NOPTRS      GET SIZE OF PRINTER ECB'S
000074 8B30 0002            00002   94           SLA   ODDREG,2
000078 4102 3170            00170   95           LA    ZEROREG,368(EVENREG,ODDREG)  GET SIZE OF CONTROL TABLE
00007C 5000 BA8C            00AA0   96           ST    ZEROREG,TABSIZE    SAVE SIZE
                                    97           GETMAIN  R,LV=(0)
000080 4510 B070            00084   98+          BAL   1,*+4 INDICATE GETMAIN
000084 0A0A                         99+          SVC   10 ISSUE GETMAIN SVC
000086 1891                        100           LR    TABBASE,PARMREG    GET ADDR OF CONTROL TABLE
000088 D201 9000 BA86 00000 00A9A  101           MVC   0(2,TABBASE),NODVCS+2  SAVE NO. OF DEVICES
00008E D701 9002 9002 00002 00002  102           XC    2(2,TABBASE),2(TABBASE) CLEAR ACTIVE NO. OF DEVICES
000094 4112 9010            00010  103           LA    PARMREG,16(EVENREG,TABBASE) GET ADDR OF ECB LIST
000098 5019 0004            00004  104           ST    PARMREG,4(TABBASE)  & SAVE IT
00009C 4111 3004            00004  105           LA    PARMREG,4(PARMREG,ODDREG) GET ADDR OF DECB
0000A0 5019 0008            00008  106           ST    PARMREG,8(TABBASE)  & SAVE IT
0000A4 4111 0028            00028  107           LA    PARMREG,40(PARMREG) GET ADDR OF BUFFER
0000A8 5019 000C            0000C  108           ST    PARMREG,12(TABBASE)  & SAVE IT
                                   109 *
0000AC 4112 315F            0015F  110           LA    PARMREG,351(EVENREG,ODDREG) GET LENGTH TO CLEAR
```

```
  LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F01OCT71   3/22/72

0000B0 9200 9010        00010        111            MVI    16(TABBASE),0         CLEAR FIRST BYTE
0000B4 4129 0010              00010  112            LA     EVENREG,16(TABBASE)  INITIALIZE ADDRESS POINTER
0000B8 5910 BA90              00AA4  113 CLRLOOP    C      PARMREG,F256          MOVE LENGTH OVER 256
0000BC 4740 B0C2              000D6  114            BL     LASTMOVE              NO, DO LAST MOVE
0000C0 D2FF 2001 2000 00001 00000    115            MVC    1(256,EVENREG),0(EVENREG) CLEAR 256 BYTES
0000C6 4122 0100              00100  116            LA     EVENREG,256(EVENREG)  INCREMENT ADDRESS POINTER
0000CA 5B10 BA90              00AA4  117            S      PARMREG,F256          DECR MOVE COUNT
0000CE 4770 B0A4              000B8  118            BNZ    CLRLOOP               REPEAT IF MORE TO DO
0000D2 47F0 B0D2              000E6  119            B      GETPTRS               GET PRINTER RLNS
                                     120 *
0000D6 0610                          121 LASTMOVE BCTR PARMREG,0             DECR COUNT FOR EXECUTE
0000D8 4410 B0CC              000E0  122            EX     PARMREG,CLRMOVE       CLEAR REST OF AREA
0000DC 47F0 B0D2              000E6  123            B      GETPTRS               GET PRINTER RLNS
                                     124 *
0000E0 D200 2001 2000 00001 00000    125 CLRMOVE    MVC    1(0,EVENREG),0(EVENREG)
                                     126 *
                                     127 *
0000E6 1B77                          128 GETPTRS    SR     LOOPREG2,LOOPREG2     INITIALIZE COUNTER
0000E8 1B88                          129            SR     INDXREG2,INDXREG2     INITIALIZE PRINTER ECB INDEX
0000EA 5970 BA88              00A9C  130 PTRLOOP    C      LOOPREG2,NOPTRS       ALL PTR RLNS OBTAINED
0000EE 47B0 B122              00136  131            BNL    FININIT               YES, COMPLETE INITIALIZATION
0000F2 1244                          132 ENDLIST    LTR    LOOPREG,LOOPREG       ANY MORE PARMS
0000F4 4780 B114              00128  133            BZ     DFLTRLN               NO, TAKE DEFAULTS
0000F8 45E0 B8FA              0090E  134            BAL    RETNREG,GETPARM
0000FC 1211                          135            LTR    PARMREG,PARMREG       RLN VALID
0000FE 4780 B0DE              000F2  136            BZ     ENDLIST               NO, IGNORE IT
000102 5910 BA84              00A98  137            C      PARMREG,NODVCS
000106 4720 B0DE              000F2  138            BH     ENDLIST
00010A 8B10 0003              00003  139 INITPTR    SLA    PARMREG,3             GET INDEX TO ENTRY
00010E 4119 1008              00008  140            LA     PARMREG,8(TABBASE,PARMREG) GET ADDR OF ENTRY FOR PTR
000112 D201 1000 BB3C 00000 00B50    141            MVC    0(2,PARMREG),MIN1     FREE PTR FORMAT
000118 4081 0002              00002  142            STH    INDXREG2,2(PARMREG)   INDEX TO ECB PTR
00011C 4188 0004              00004  143            LA     INDXREG2,4(INDXREG2)  INCR TO NEXT ECB
000120 4177 0001              00001  144            LA     LOOPREG2,1(LOOPREG2)  INCR PRINTER COUNTER
000124 47F0 B0D6              000EA  145            B      PTRLOOP
                                     146 *
                                     147 *
000128 1277                          148 DFLTRLN    LTR    LOOPREG2,LOOPREG2     NO PTR RLNS PROCESSED
00012A 4770 B122              00136  149            BNZ    FININIT               NO, COMPLETE INITIALIZATION
00012E 4110 0001              00001  150            LA     PARMREG,1             YES, DEFAULT RLN = 1
000132 47F0 B0F6              0010A  151            B      INITPTR
                                     152 *
                                     153 *
000136 5810 BA84              00A98  154 FININIT    L      PARMREG,NODVCS        GET NUMBER OF DISPLAYS
00013A 5B10 BA88              00A9C  155            S      PARMREG,NOPTRS
00013E 1841                          156            LR     LOOPREG,PARMREG       SAVE LOOP COUNTER
000140 5C00 BA94              00AA8  157            M      ZEROREG,DATASZ        GET SIZE OF DATA AREAS
000144 1801                          158            LR     ZEROREG,PARMREG
000146 5000 BA98              00AAC  159            ST     ZEROREG,DATBLKSZ      SAVE DATA AREA SIZE
                                     160            GETMAIN  R,LV=(0)            GET CORE FOR DATA AREAS
00014A 4510 B13A              0014E  161+           BAL    1,*+4 INDICATE GETMAIN
00014E 0A0A                          162+           SVC    10 ISSUE GETMAIN SVC
000150 5010 BA9C              00AB0  163            ST     PARMREG,DATBLKAD      SAVE DATA AREA ADDR
000154 1851                          164            LR     INDXREG,PARMREG       GET ADDRESS OF AREA
000156 4189 0010              00010  165            LA     INDXREG2,16(TABBASE)  GET ADDR OF ENTRY
```

```
  LOC  OBJECT CODE      ADDR1 ADDR2  STMT    SOURCE STATEMENT                                                    F01OCT71   3/22/72

00015A 1B66                          166           SR    CNTREG,CNTREG        PRINTER RLN
00015C 4170 0001              00001  167           LA    LOOPREG2,1           RLN COUNTER
                                     168 *
000160 D501 8000 BB3C 00000 00B50    169 INITLOOP CLC    0(2,INDXREG2),MIN1   PRINTER ENTRY
000166 4780 B192              001A6  170           BE    PTRINIT              YES, GET RLN
00016A D781 5000 5000 00000 00000    171           XC    0(130,INDXREG),0(INDXREG)  CLEAR DATA AREA
000170 4068 0002              00002  172           STH   CNTREG,2(INDXREG2)   STORE PRINTER RLN FOR THIS DISPLAY
000174 5058 0004              00004  173           ST    INDXREG,4(INDXREG2)  STORE ADDR OF DATA AREA
000178 5A50 BA94              00AA8  174           A     INDXREG,DATASZ       INCR DATA AREA PTR
00017C 4188 0008              00008  175 INCRPTRS LA     INDXREG2,8(INDXREG2) INCR ENTRY PTR
000180 4177 0001              00001  176           LA    LOOPREG2,1(LOOPREG2) INCR RLN
000184 4640 B14C              00160  177           BCT   LOOPREG,INITLOOP     BR IF MORE TO DO
000188 5819 0004              00004  178           L     PARMREG,4(TABBASE)   GET ADDR OF ECB LIST
00018C 5850 BA88              00A9C  179           L     INDXREG,NOPTRS       GET INDEX PAST PRINTER ECB
000190 8B50 0002              00002  180           SLA   INDXREG,2
000194 4111 5000              00000  181           LA    PARMREG,0(PARMREG,INDXREG)  GET ADDR OF DISP ENTRY
000198 9280 1000        00000        182           MVI   0(PARMREG),X'80'     SET VL FLAG
00019C D202 1001 9009 00001 00009    183           MVC   1(3,PARMREG),9(TABBASE)  MOVE IN ADDR OF ECB
0001A2 47F0 B198              001AC  184           B     IOINIT
0001A6 1867                          185 PTRINIT  LR     CNTREG,LOOPREG2      GET RLN OF PRINTER
0001A8 47F0 B168              0017C  186           B     INCRPTRS
                                     187 IOINIT   OPEN   (SYSPRINT,OUTPUT,DD3270)
0001AC                               188+         CNOP   0,4
0001AC 4510 B1A4              001B8  189+IOINIT   BAL    1,*+12 LOAD REG1 W/LIST ADDR.
0001B0 0F                            190+         DC     AL1(15) OPTION BYTE
0001B1 000ACC                        191+         DC     AL3(SYSPRINT) DCB ADDRESS
0001B4 80                            192+         DC     AL1(128) OPTION BYTE
0001B5 000B18                        193+         DC     AL3(DD3270) DCB ADDRESS
0001B8 0A13                          194+         SVC    19 ISSUE OPEN SVC
                                     195          WTO    'LOCAL 3270 SAMPLE PROGRAM RUNNING'
0001BA 0700                          196+         CNOP   0,4
0001BC 4510 B1D2              001E6  197+         BAL    1,IHB0005A BRANCH AROUND MESSAGE
0001C0 0025                          198+         DC     AL2(37) TEXT LENGTH
0001C2 0000                          199+         DC     B'0000000000000000' MCS FLAGS
0001C4 D3D6C3C1D340F3F2              200+         DC     C'LOCAL 3270 SAMPLE PROGRAM RUNNING'
0001E6                               201+IHB0005A DS     0H
0001E6 0A23                          202+         SVC    35
0001E8 5840 BA84              00A98  203           L     LOOPREG,NODVCS       GET LOOP CTR
0001EC 5879 0008              00008  204           L     DECBREG,8(TABBASE)   GET ADDR OF DECB
0001F0 5889 000C              0000C  205           L     BUFREG,12(TABBASE)   GET ADDR OF BUFFER
0001F4 4159 0010              00010  206           LA    INDXREG,16(TABBASE)  GET ADDR OF FIRST TERM ENTRY
0001F8 4160 0001              00001  207           LA    RLNREG,1             INITIALIZE RLN REG
                                     208 *
0001FC 41F0 B6E8              006FC  209 IOLOOP    LA    EPREG,WRTFMT1        GET ADDR OF FIRST WRITE RTN
000200 D501 5000 BB3C 00000 00B50    210           CLC   0(2,INDXREG),MIN1    PRINTER ENTRY
000206 4780 B20A              0021E  211           BE    UPCOUNT              YES, DON'T DO WRITE
00020A 1816                          212           LR    PARMREG,RLNREG       PUT RLN IN REG 1
00020C 05EF                          213           BALR  RETNREG,EPREG        WRITE FIRST MSG
00020E 12FF                          214           LTR   EPREG,EPREG          SUCCESSFUL
000210 4780 B20A              0021E  215           BZ    UPCOUNT              YES, TRY NEXT
000214 D701 5000 5000 00000 00000    216           XC    0(2,INDXREG),0(INDXREG) MARK TERMINAL INACTIVE
00021A 47F0 B216              0022A  217           B     IOLPCNT
                                     218 *
00021E 4829 0002              00002  219 UPCOUNT   LH    EVENREG,2(TABBASE)   INCREMENT
000222 4122 0001              00001  220           LA    EVENREG,1(EVENREG)         COUNT OF ACTIVE
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F01OCT71   3/22/72

000226 4029 0002            00002   221            STH  EVENREG,2(TABBASE)              DEVICES
00022A 4166 0001            00001   222 IOLPCNT    LA   RLNREG,1(RLNREG)     INCR RLN
00022E 4155 0008            00008   223            LA   INDXREG,8(INDXREG)   INCR TO NEXT ENTRY
000232 4640 B1E8            001FC   224            BCT  LOOPREG,IOLOOP       LOOP IF MORE TO DO
                                    225 *
000236 4160 0001            00001   226            LA   CNTREG,1             INIT RLN
00023A 4159 0010            00010   227            LA   INDXREG,16(TABBASE)  POINTER TO FIRST ENTRY
00023E 5840 BA84            00A98   228            L    LOOPREG,NODVCS       LOOP LIMIT
000242 D501 5000 BB3C 00000 00B50   229 FINDFRST   CLC  0(2,INDXREG),MIN1    PRINTER
000248 4770 B248            0025C   230            BNE  DSPLY1               NO, SAVE RLN
00024C 4166 0001            00001   231            LA   CNTREG,1(CNTREG)     INCR RLN
000250 4155 0008            00008   232            LA   INDXREG,8(INDXREG)   INCR TO NEXT ENTRY
000254 4640 B22E            00242   233            BCT  LOOPREG,FINDFRST
000258 4160 0001            00001   234            LA   CNTREG,1
00025C 5060 BAA0            00AB4   235 DSPLY1     ST   CNTREG,READRLN       SAVE RLN FOR READS
                                    236 *
000260 5860 BAA0            00AB4   237 READ       L    RLNREG,READRLN       GET READ RLN
000264 D703 7000 7000 00000 00000   238            XC   0(4,DECBREG),0(DECBREG)
                                    239            READ (DECBREG),TI,DD3270,(BUFREG),308,,(RLNREG),MF=E
00026A 1817                         240+           LR   1,DECBREG LOAD DECB ADDRESS
00026C 9404 1004      00004         241+           NI   4(1),4
000270 9201 1005      00005         242+           MVI  5(1),1
000274 41E0 0134            00134   243+           LA   14,308(0,0) 000B
000278 40E0 1006            00006   244+           STH  14,6(0,1) STORE LENGTH       000B
00027C 41E0 BB04            00B18   245+           LA   14,DD3270
000280 50E0 1008            00008   246+           ST   14,8(0,1) STORE DCB ADDRESS
000284 5080 100C            0000C   247+           ST   BUFREG,12(0,1) STORE AREA ADDR   000B
000288 4060 1018            00018   248+           STH  RLNREG,24(0,1) STORE LINE NUMBER
00028C 58F0 BB34            00B48   249+           L    15,DD3270+48 LOAD RDWRT ROUT ADDR
000290 94F7 1004      00004         250+           NI   4(1),X'F7' A38557
000294 05EF                         251+           BALR 14,15 A38557
000296 12FF                         252            LTR  EPREG,EPREG          OK
000298 4770 B8EC            00900   253            BNZ  ERRABEND             NO, ABEND
00029C 5839 0004            00004   254 TWAIT      L    ODDREG,4(TABBASE)    GET ADDR OF ECB LIST
                                    255            TWAIT (EVENREG),ECBLIST=(ODDREG)  WAIT FOR OPERATION TO END
0002A0                              256+           DS   0H
0002A0                              257+IECA0008   EQU  *
0002A0 4113 0000            00000   258+           LA   1,0((ODDREG)) 000A
0002A4 1801                         259+           LR   0,1 SAVE PARAMETER LIST ADDRESS.
0002A6 58F1 0000            00000   260+IECB0008   L    15,0(1) LOAD ECB ADDRESS.
0002AA 9140 F000      00000         261+           TM   0(15),X'40' IS COMPLETE BIT ON?
0002AE 4770 B2C2            002D6   262+           BNZ  IECC0008
0002B2 9180 1000      00000         263+           TM   0(1),X'80' IF NOT, TEST FOR END OF LIST.
0002B6 4710 B2AE            002C2   264+           BO   IECD0008
0002BA 4111 0004            00004   265+           LA   1,4(1) STEP TO NEXT ENTRY IN LIST.
0002BE 47F0 B292            002A6   266+           B    IECB0008
0002C2 1810                         267+IECD0008   LR   1,0 RESTORE PARAMETER LIST REGISTER.
0002C4 4100 0001            00001   268+           LA   0,1 LOAD WAIT COUNT.
0002C8 4110 1000            00000   269+           LA   1,0(0,1) CLEAR HIGH BYTE OF REG
0002CC 1800                         270+           LR   0,0
0002CE 1311                         271+           LCR  1,1 INDICATE ECBLIST USED
0002D0 0A01                         272+           SVC  1 LINK TO WAIT ROUTINE
0002D2 47F0 B28C            002A0   273+           B    IECA0008
0002D6 412F 0000            00000   274+IECC0008   LA   (EVENREG),0(15) IF ON, LOAD SPECIFIED REGISTER WITH
                                    275+*                                ADDRESS OF COMPLETED ECB
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  F01OCT71   3/22/72

0002DA 18F1                         276+          LR    15,1
0002DC 1BF0                         277+          SR    15,0
                                    278 *
0002DE 9501 2005       00005        279           CLI   TYPE(EVENREG),X'01' OPERATION = READ INITIAL
0002E2 4770 B760             00774  280           BNE   PRNTREND            NO, PRINTER OPERATION
0002E6 9544 2000       00000        281           CLI   0(EVENREG),X'44'    INTERCEPTED
0002EA 4780 B24C             00260  282           BE    READ                YES, RETRY READ
0002EE 957F 2000       00000        283           CLI   0(EVENREG),X'7F'    SUCCESSFUL
0002F2 4770 B8EC             00900  284           BNE   ERRABEND            NO, END
0002F6 5812 0024             00024  285           L     PARMREG,POLPT(EVENREG) GET RESPONDING RLN
0002FA 1831                         286           LR    ODDREG,PARMREG
0002FC 8B30 0003             00003  287           SLA   ODDREG,3            GET INDEX TO TERM ENTRY
000300 4869 3008             00008  288           LH    CNTREG,8(TABBASE,ODDREG)  GET FORMAT NO.
000304 8B60 0002             00002  289           SLA   CNTREG,2            MULTIPLY BY 4
000308 47F6 B2F8             0030C  290           B     READRTN(CNTREG)
00030C 47F0 B8EC             00900  291 READRTN   B     ERRABEND            RLN = INACTIVE DEVICE
000310 47F0 B30C             00320  292           B     READFMT1            FORMAT 1 ON SCREEN
000314 47F0 B5C2             005D6  293           B     READFMT2            FORMAT 2 ON SCREEN
000318 47F0 B6B8             006CC  294           B     READFMT3            FORMAT 3 ON SCREEN
00031C 47F0 B24C             00260  295           B     READ                FORMAT 4 ON SCREEN - IGNORE IT
                                    296 *
                                    297 *
000320 956E 8000       00000        298 READFMT1  CLI   0(BUFREG),X'6E'     INTERRUPT IS PA2 KEY
000324 4770 B40A             0041E  299           BNE   CLR1                NO, TEST FOR CLEAR KEY
                                    300 *
000328 1861                         301 WRTFMT4   LR    RLNREG,PARMREG      SAVE RLN
000328                              302 ENDMSG    EQU   WRTFMT4             RETURN = REESTABLISH READ
00032A 8B10 0003             00003  303           SLA   PARMREG,3           CONVERT RLN TO TABLE INDEX
00032E 4120 0004             00004  304           LA    EVENREG,4           FORMAT ID
000332 4029 1008             00008  305           STH   EVENREG,8(TABBASE,PARMREG)  STORE ID IN TABLE
                                    306           CHGNTRY  DD3270,ATTLST,(RLNREG),,SKIP     DEACTIVATE TERMINAL
000336                              307+          DS    0H
000336 4110 BB04             00B18  308+          LA    1,DD3270
00033A 58F1 002C             0002C  309+          L     15,44(1) GET DEB ADDR
00033E 58EF 0018             00018  310+          L     14,24(15) GET DCB ADDR FROM DEB
000342 41EE 0000             00000  311+          LA    14,0(14) CLEAR HIGH-ORDER BYTE
000346 4111 0000             00000  312+          LA    1,0(1) CLEAR HIGH-ORDER BYTE
00034A 191E                         313+          CR    1,14 DCB-DEB LOOP COMPLETE
00034C 4770 B362             00376  314+          BNE   IECA0010 NO, GIVE RETURN CODE = 8
000350 181F                         315+          LR    1,15 DEB ADDR TO REG 1
000352 41F0 0006             00006  316+          LA    15,(RLNREG) GET RLN
000356 1BEE                         317+          SR    14,14
000358 43E1 0010             00010  318+          IC    14,16(1) GET NUMBER OF EXTENTS
00035C 19FE                         319+          CR    15,14 IS RLN VALID
00035E 4720 B362             00376  320+          BH    IECA0010 NO, GIVE RETURN CODE = 8
000362 8BF0 0002             00002  321+          SLA   15,2 MULTIPLY RLN BY 4
000366 581F 101C             0001C  322+          L     1,28(15,1) GET ADDR OF UCB FROM DEB
00036A 1B00                         323+          SR    0,0
00036C 41F0 0002             00002  324+          LA    15,2 FUNCTION IS SKIP
000370 0A74                         325+          SVC   116 INVOKE ESR
000372 47F0 B366             0037A  326+          B     IECB0010 BRANCH TO EXIT
000376 41F0 0008             00008  327+IECA0010 LA    15,8 SET RETURN CODE OF 8
00037A                              328+IECB0010 EQU   *
                                    329 *                               FOR READ INITIAL
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                 F01OCT71    3/22/72

00037A D703 7000 7000 00000 00000   330 WRITE4   XC    0(4,DECBREG),0(DECBREG)   CLEAR ECB
                                    331          WRITE (DECBREG),TS,DD3270,FORMAT4,FMT4SZ,,(RLNREG),MF=E
000380 1817                         332+         LR    1,DECBREG LOAD DECB ADDRESS
000382 9404 1004      00004         333+         NI    4(1),4
000386 920E 1005      00005         334+         MVI   5(1),14
00038A 47F0 B37C            00390   335+         B     *+6 000B
00038E 0024                         336+         DC    AL2(FMT4SZ) 000B
000390 D201 1006 B37A 00006 0038E   337+         MVC   6(2,1),*-2 MOVE IN LENGTH    000B
000396 41E0 BB04            00B18   338+         LA    14,DD3270
00039A 50E0 1008            00008   339+         ST    14,8(0,1) STORE DCB ADDRESS
00039E 41E0 BE76            00E8A   340+         LA    14,FORMAT4 000B
0003A2 50E0 100C            0000C   341+         ST    14,12(0,1) STORE AREA ADDR   000B
0003A6 4060 1018            00018   342+         STH   RLNREG,24(0,1) STORE LINE NUMBER
0003AA 58F0 BB34            00B48   343+         L     15,DD3270+48 LOAD RDWRT ROUT ADDR
0003AE 94F7 1004      00004         344+         NI    4(1),X'F7' A38557
0003B2 05EF                         345+         BALR  14,15 A38557
0003B4 12FF                         346          LTR   EPREG,EPREG          END IF RC
0003B6 4770 B8EC            00900   347          BNZ   ERRABEND                     NON-ZERO
                                    348          WAIT  ECB=(DECBREG)
0003BA 4110 7000            00000   349+         LA    1,0(0,DECBREG) CLEAR HIGH BYTE OF REG
0003BE 4100 0001            00001   350+         LA    0,1(0,0) COUNT OMITTED,1 USED
0003C2 0A01                         351+         SVC   1 LINK TO WAIT ROUTINE
0003C4 957F 7000      00000         352          CLI   0(DECBREG),X'7F'     SATISFACTORY COMPLETION
0003C8 4780 B3C4            003D8   353          BE    QUIESCE              UPDATE ACTIVE TERM COUNT
0003CC 9544 7000      00000         354          CLI   0(DECBREG),X'44'     INTERCEPT
0003D0 4780 B366            0037A   355          BE    WRITE4               YES, RETRY
0003D4 47F0 B8EC            00900   356          B     ERRABEND             NO, END
                                    357 *
                                    358 *
0003D8 4829 0002            00002   359 QUIESCE  LH    EVENREG,2(TABBASE)   GET NO. OF ACTIVE TERMS
0003DC 4620 B402            00416   360          BCT   EVENREG,QCNTSTR      DECR COUNT
                                    361          CLOSE (SYSPRINT,,DD3270)   IF ZERO, CLOSE DCB'S
0003E0                              362+         CNOP  0,4
0003E0 4510 B3D8            003EC   363+         BAL   1,*+12 BRANCH AROUND LIST
0003E4 00                           364+         DC    AL1(0) OPTION BYTE
0003E5 000ACC                       365+         DC    AL3(SYSPRINT) DCB ADDRESS
0003E8 80                           366+         DC    AL1(128) OPTION BYTE
0003E9 000B18                       367+         DC    AL3(DD3270) DCB ADDRESS
0003EC 0A14                         368+         SVC   20 ISSUE CLOSE SVC
0003EE 5810 BA9C            00AB0   369          L     PARMREG,DATBLKAD     LOCATION OF DATA AREAS
0003F2 5800 BA98            00AAC   370          L     ZEROREG,DATBLKSZ     SIZE OF DATA AREAS
                                    371          FREEMAIN  R,LV=(0),A=(1)   FREE DATA AREAS
0003F6 4111 0000            00000   372+         LA    1,0(1) CLEAR THE HIGH ORDER BYTE
0003FA 0A0A                         373+         SVC   10 ISSUE FREEMAIN SVC
0003FC 1819                         374          LR    PARMREG,TABBASE      LOCATION OF CONTROL TABLE
0003FE 5800 BA8C            00AA0   375          L     ZEROREG,TABSIZE      SIZE OF CONTROL TABLE
                                    376          FREEMAIN  R,LV=(0),A=(1)   FREE CONTROL TABLE
000402 4111 0000            00000   377+         LA    1,0(1) CLEAR THE HIGH ORDER BYTE
000406 0A0A                         378+         SVC   10 ISSUE FREEMAIN SVC
000408 58D0 BA3C            00A50   379          L     SAVEREG,SAVE+4
                                    380          RETURN  (14,12),T          RETURN TO CALLING PROGRAM
00040C 98EC D00C            0000C   381+         LM    14,12,12(13) RESTORE THE REGISTERS
000410 92FF D00C      0000C         382+         MVI   12(13),X'FF' SET RETURN INDICATION
000414 07FE                         383+         BR    14 RETURN
                                    384 *
```

```
  LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                F01OCT71    3/22/72

000416 4029 0002              00002   385 QCNTSTR  STH   EVENREG,2(TABBASE)  SAVE UPDATE ACTIVE TERM CNT
00041A 47F0 B24C              00260   386          B     READ                SET UP READ
                                      387 *
                                      388 *
00041E 956D 8000        00000         389 CLR1     CLI   0(BUFREG),X'6D'     CLEAR KEY
000422 4770 B47E              00492   390          BNE   ENT1                NO, TEST FOR ENTER KEY
                                      391 *
000426 1861                           392 WRTFMT3  LR    RLNREG,PARMREG      SAVE RLN
000426                                393 CNTRLMSG EQU   WRTFMT3
                                      394 *                                  WRITE CONTROL OPTIONS
000428 8B10 0003              00003   395          SLA   PARMREG,3           CONVERT RLN TO TABLE INDEX
00042C 4120 0003              00003   396          LA    EVENREG,3           FORMAT ID
000430 4029 1008              00008   397          STH   EVENREG,8(TABBASE,PARMREG)  STORE ID IN TABLE
000434 D703 7000 7000 00000 00000     398 WRITE3   XC    0(4,DECBREG),0(DECBREG)  CLEAR ECB
                                      399          WRITE (DECBREG),TS,DD3270,FORMAT3,FMT3SZ,,(RLNREG),MF=E
00043A 1817                           400+         LR    1,DECBREG LOAD DECB ADDRESS
00043C 9404 1004        00004         401+         NI    4(1),4
000440 920E 1005        00005         402+         MVI   5(1),14
000444 47F0 B436              0044A   403+         B     *+6 000B
000448 0148                           404+         DC    AL2(FMT3SZ) 000B
00044A D201 1006 B434 00006 00448     405+         MVC   6(2,1),*-2 MOVE IN LENGTH     000B
000450 41E0 BB04              00B18   406+         LA    14,DD3270
000454 50E0 1008              00008   407+         ST    14,8(0,1) STORE DCB ADDRESS
000458 41E0 BD2E              00D42   408+         LA    14,FORMAT3 000B
00045C 50E0 100C              0000C   409+         ST    14,12(0,1) STORE AREA ADDR    000B
000460 4060 1018              00018   410+         STH   RLNREG,24(0,1) STORE LINE NUMBER
000464 58F0 BB34              00B48   411+         L     15,DD3270+48 LOAD RDWRT ROUT ADDR
000468 94F7 1004        00004         412+         NI    4(1),X'F7' A38557
00046C 05EF                           413+         BALR  14,15 A38557
00046E 12FF                           414          LTR   EPREG,EPREG         END IF RC
000470 4770 B8EC              00900   415          BNZ   ERRABEND                 NON-ZERO
                                      416          WAIT  ECB=(DECBREG)       WAIT FOR COMPLETION
000474 4110 7000              00000   417+         LA    1,0(0,DECBREG) CLEAR HIGH BYTE OF REG
000478 4100 0001              00001   418+         LA    0,1(0,0) COUNT OMITTED,1 USED
00047C 0A01                           419+         SVC   1 LINK TO WAIT ROUTINE
00047E 957F 7000        00000         420          CLI   0(DECBREG),X'7F'    SATISFACTORY COMPLETION
000482 4780 B24C              00260   421          BE    READ                YES, SET UP READ
000486 9544 7000        00000         422          CLI   0(DECBREG),X'44'    INTERCEPT
00048A 4780 B420              00434   423          BE    WRITE3              YES, RETRY OPERATION
00048E 47F0 B8EC              00900   424          B     ERRABEND            NO, END
                                      425 *
000492 957D 8000        00000         426 ENT1     CLI   0(BUFREG),X'7D'     ENTER KEY
000496 4770 B24C              00260   427          BNE   READ                NO, REESTABLISH READ
00049A 5839 300C              0000C   428          L     ODDREG,12(TABBASE,ODDREG) GET ADDR OF DATA AREA
00049E 4158 0003              00003   429          LA    INDXREG,3(BUFREG)   INDEX PAST AID & CURSOR ADDR
0004A2 4140 0134              00134   430          LA    LOOPREG,308
0004A6 4B42 0012              00012   431          SH    LOOPREG,COUNT(EVENREG) GET NUMBER OF BYTES READ
0004AA 5B40 BAA8              00ABC   432          S     LOOPREG,THREE       ADJUST FOR LENGTH OF AID & ADDR
0004AE 1244                           433          LTR   LOOPREG,LOOPREG
0004B0 47D0 B52C              00540   434          BNP   WRTRSPNS            NOTHING ENTERED
                                      435 *
0004B4 D502 5000 BB3F 00000 00B53     436          CLC   0(3,INDXREG),FLD1   FIRST FIELD
0004BA 4770 B4BA              004CE   437          BNE   FLD2CHK             NO, CHECK FOR SECOND
0004BE D720 3000 3000 00000 00000     438          XC    0(33,ODDREG),0(ODDREG)   CLEAR FIELD IN DATA AREA
0004C4 45E0 B97A              0098E   439          BAL   RETNREG,GETFIELD    FIND FIELD DELIMITERS
```

```
  LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  F01OCT71    3/22/72

0004C8 1244                          440           LTR    LOOPREG,LOOPREG       END OF DATA
0004CA 47D0 B52C               00540 441           BNP    WRTRSPNS              YES, WRITE RESPONSE
                                     442 *
0004CE 4133 0022               00022 443 FLD2CHK   LA     ODDREG,34(ODDREG)     INCR TO NEXT FIELD IN DATA AREA
0004D2 D502 5000 BB42 00000 00B56    444           CLC    0(3,INDXREG),FLD2     SECOND FIELD
0004D8 4770 B4D8               004EC 445           BNE    FLD3CHK               NO, CHECK FOR THIRD FIELD
0004DC D720 3000 3000 00000 00000    446           XC     0(33,ODDREG),0(ODDREG)    CLEAR FIELD IN DATA AREA
0004E2 45E0 B97A               0098E 447           BAL    RETNREG,GETFIELD      GET FIELD
0004E6 1244                          448           LTR    LOOPREG,LOOPREG       END OF DATA
0004E8 47D0 B52C               00540 449           BNP    WRTRSPNS              YES, WRITE RESPONSE
                                     450 *
0004EC 4133 0022               00022 451 FLD3CHK   LA     ODDREG,34(ODDREG)     INCR TO THIRD FIELD IN DATA AREA
0004F0 D502 5000 BB45 00000 00B59    452           CLC    0(3,INDXREG),FLD3     THIRD FIELD
0004F6 4770 B4F6               0050A 453           BNE    FLD4CHK               NO, CHECK FOR FOURTH FIELD
0004FA D720 3000 3000 00000 00000    454           XC     0(33,ODDREG),0(ODDREG)    CLEAR FIELD
000500 45E0 B97A               0098E 455           BAL    RETNREG,GETFIELD      GET FIELD
000504 1244                          456           LTR    LOOPREG,LOOPREG       END OF DATA
000506 47D0 B52C               00540 457           BNP    WRTRSPNS              YES, WRITE RESPONSE
                                     458 *
00050A 4133 0022               00022 459 FLD4CHK   LA     ODDREG,34(ODDREG)     INCR TO FOURTH FIELD
00050E D502 5000 BB48 00000 00B5C    460           CLC    0(3,INDXREG),FLD4     FOURTH FIELD
000514 4770 B514               00528 461           BNE    FLD5CHK               NO, CHECK FOR LAST FIELD
000518 D714 3000 3000 00000 00000    462           XC     0(21,ODDREG),0(ODDREG)    CLEAR FIELD
00051E 45E0 B97A               0098E 463           BAL    RETNREG,GETFIELD      GET FIELD
000522 1244                          464           LTR    LOOPREG,LOOPREG       END OF DATA
000524 47D0 B52C               00540 465           BNP    WRTRSPNS              YES, WRITE RESPONSE
                                     466 *
000528 4133 0016               00016 467 FLD5CHK   LA     ODDREG,22(ODDREG)     INCR TO LAST FIELD
00052C D502 5000 BB4B 00000 00B5F    468           CLC    0(3,INDXREG),FLD5     FIFTH FIELD
000532 4770 B52C               00540 469           BNE    WRTRSPNS              NO, WRITE RESPONSE
000536 D704 3000 3000 00000 00000    470           XC     0(5,ODDREG),0(ODDREG) CLEAR FIELD
00053C 45E0 B97A               0098E 471           BAL    RETNREG,GETFIELD      GET FIELD
                                     472 *
                                     473 *
000540 1861                          474 WRTFMT2   LR     RLNREG,PARMREG        SAVE RLN
000540                               475 WRTRSPNS EQU     WRTFMT2               RETURN = REESTABLISH READ
                                     476 *                                      WRITE FORMAT 2
000542 D2FF 8000 BBFC 00000 00C10    477           MVC    0(256,BUFREG),FORMAT2  MOVE BASE MESSAGE
000548 D231 8100 BCFC 00100 00D10    478           MVC    256(FMT2SZ-256,BUFREG),FORMAT2+256  INTO BUFFER
00054E 8B10 0003               00003 479           SLA    PARMREG,3             CONVERT RLN TO TABLE INDEX
000552 4120 0002               00002 480           LA     EVENREG,2             GET FORMAT ID
000556 4029 1008               00008 481           STH    EVENREG,8(TABBASE,PARMREG)  & STORE IN TABLE ENTRY
00055A 5839 100C               0000C 482           L      ODDREG,12(TABBASE,PARMREG)  GET DATA AREA ADDR
00055E D220 8028 3000 00028 00000    483           MVC    FMT2FLD1(33,BUFREG),0(ODDREG)       MOVE DATA
000564 D220 8052 3022 00052 00022    484           MVC    FMT2FLD2(33,BUFREG),34(ODDREG)      FROM
00056A D220 807C 3044 0007C 00044    485           MVC    FMT2FLD3(33,BUFREG),68(ODDREG)      DATA AREA
000570 D214 80A7 3066 000A7 00066    486           MVC    FMT2FLD4(21,BUFREG),102(ODDREG)     TO
000576 D204 80C4 307C 000C4 0007C    487           MVC    FMT2FLD5(5,BUFREG),124(ODDREG)      BUFFER
00057C D703 7000 7000 00000 00000    488 WRITE2    XC     0(4,DECBREG),0(DECBREG)  CLEAR ECB
                                     489           WRITE  (DECBREG),TS,DD3270,(BUFREG),FMT2SZ,,(RLNREG),MF=E
000582 1817                          490+          LR     1,DECBREG LOAD DECB ADDRESS
000584 9404 1004      00004          491+          NI     4(1),4
000588 920E 1005      00005          492+          MVI    5(1),14
00058C 47F0 B57E               00592 493+          B      *+6 000B
000590 0132                          494+          DC     AL2(FMT2SZ) 000B
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  F01OCT71   3/22/72

000592 D201 1006 B57C 00006 00590   495+           MVC   6(2,1),*-2 MOVE IN LENGTH     000B
000598 41E0 BB04            00B18   496+           LA    14,DD3270
00059C 50E0 1008            00008   497+           ST    14,8(0,1) STORE DCB ADDRESS
0005A0 5080 100C            0000C   498+           ST    BUFREG,12(0,1) STORE AREA ADDR   000B
0005A4 4060 1018            00018   499+           STH   RLNREG,24(0,1) STORE LINE NUMBER
0005A8 58F0 BB34            00B48   500+           L     15,DD3270+48 LOAD RDWRT ROUT ADDR
0005AC 94F7 1004      00004         501+           NI    4(1),X'F7' A38557
0005B0 05EF                         502+           BALR  14,15 A38557
0005B2 12FF                         503            LTR   EPREG,EPREG         END IF RC
0005B4 4770 B8EC            00900   504            BNZ   ERRABEND                    NON-ZERO
                                    505            WAIT  ECB=(DECBREG)       WAIT FOR OPERATION TO END
0005B8 4110 7000            00000   506+           LA    1,0(0,DECBREG) CLEAR HIGH BYTE OF REG
0005BC 4100 0001            00001   507+           LA    0,1(0,0) COUNT OMITTED,1 USED
0005C0 0A01                         508+           SVC   1 LINK TO WAIT ROUTINE
0005C2 957F 7000      00000         509            CLI   0(DECBREG),X'7F'    COMPLETION SATISFACTORY
0005C6 4780 B24C            00260   510            BE    READ                YES, SET UP READ
0005CA 9544 7000      00000         511            CLI   0(DECBREG),X'44'    INTERCEPT
0005CE 4780 B568            0057C   512            BE    WRITE2              YES, RETRY
0005D2 47F0 B8EC            00900   513            B     ERRABEND            NO, END
                                    514 *
0005D6 956C 8000      00000         515 READFMT2 CLI   0(BUFREG),X'6C'     PA1 KEY
0005DA 4770 B30C            00320   516            BNE   READFMT1            NO, GO TO READ FORMAT 1 LOGIC
0005DE 1821                         517            LR    EVENREG,PARMREG     GET RLN OF DISPLAY
0005E0 8B20 0003            00003   518            SLA   EVENREG,3           GET INDEX TO ENTRY
0005E4 4832 900A            0000A   519            LH    ODDREG,10(EVENREG,TABBASE)  GET RLN OF ASSIGNED PTR
0005E8 1233                         520            LTR   ODDREG,ODDREG       PRINTER ASSIGNED
0005EA 4780 B682            00696   521            BZ    SYSOUT              NO, USE SYSOUT
0005EE 1863                         522            LR    RLNREG,ODDREG       SAVE RLN
0005F0 8B30 0003            00003   523            SLA   ODDREG,3            GET INDEX TO ENTRY OF PRINTER
0005F4 4100 00B8            000B8   524            LA    ZEROREG,184
                                    525            GETMAIN  R,LV=(0)         GET CORE FOR PRINTER REQUEST BLOCK
0005F8 4510 B5E8            005FC   526+           BAL   1,*+4 INDICATE GETMAIN
0005FC 0A0A                         527+           SVC   10 ISSUE GETMAIN SVC
0005FE D7B7 1000 1000 00000 00000   528            XC    0(184,PARMREG),0(PARMREG) CLEAR IT
000604 4141 002C            0002C   529            LA    LOOPREG,44(PARMREG) GET ADDR OF PRINT BUFFER
000608 5852 900C            0000C   530            L     INDXREG,12(EVENREG,TABBASE)    GET ADDR OF DATA AREA
00060C 92C8 4000      00000         531            MVI   0(LOOPREG),X'C8'    STORE WCC IN BUFFER
000610 D280 4001 5000 00001 00000   532            MVC   1(129,LOOPREG),0(INDXREG) MOVE IN DATA
000616 9215 4022      00022         533            MVI   34(LOOPREG),X'15'   NEW LINE
00061A 9215 4044      00044         534            MVI   68(LOOPREG),X'15'   NEW LINE
00061E 9215 4066      00066         535            MVI   102(LOOPREG),X'15'  NEW LINE
000622 9240 407C      0007C         536            MVI   124(LOOPREG),C' '   BLANK
000626 9219 4082      00082         537            MVI   130(LOOPREG),X'19'  END OF MESSAGE
                                    538 *
00062A 4809 3008            00008   539            LH    ZEROREG,8(TABBASE,ODDREG)  GET CURRENT PRINTER FORMAT
00062E 4900 BB3C            00B50   540            CH    ZEROREG,MIN1        PRINTER BUSY
000632 4780 B638            0064C   541            BE    STRTPRTR            NO, START OPERATION
000636 5859 300C            0000C   542            L     INDXREG,12(TABBASE,ODDREG)  GET POINTER TO LAST REQ
00063A 5015 0000            00000   543            ST    PARMREG,0(INDXREG)  STORE POINTER
00063E 5019 300C            0000C   544            ST    PARMREG,12(TABBASE,ODDREG)  UPDATE POINTER TO CURR REQ
000642 1812                         545 RSTRKYBD LR    PARMREG,EVENREG     RECOVER RLN OF DISPLAY
000644 8A10 0003            00003   546            SRA   PARMREG,3           FROM TABLE INDEX
000648 47F0 B52C            00540   547            B     WRTRSPNS            GO TO UNLOCK KEYBOARD
                                    548 *
00064C 5019 300C            0000C   549 STRTPRTR ST    PARMREG,12(TABBASE,ODDREG)  STORE POINTER TO REQ
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT                                              F01OCT71    3/22/72

000650 4111 0004             00004   550            LA    PARMREG,4(PARMREG)  GET ADDRESS OF DECB
000654 5859 0004             00004   551            L     INDXREG,4(TABBASE)  GET ADDRESS OF ECB LIST
000658 4A59 300A             0000A   552            AH    INDXREG,10(TABBASE,ODDREG)  ADD INDEX TO PRTR'S PTR
00065C 5015 0000             00000   553            ST    PARMREG,0(INDXREG)  STORE ECB ADDR IN LIST
000660 0600                          554            BCTR  ZEROREG,0           INDICATE
000662 4009 3008             00008   555            STH   ZEROREG,8(TABBASE,ODDREG)  PRINTER BUSY
                                     556            WRITE (PARMREG),TS,DD3270,(LOOPREG),131,,(RLNREG),MF=E
000666 1811                          557+           LR    1,PARMREG LOAD DECB ADDRESS
000668 9404 1004       00004         558+           NI    4(1),4
00066C 920E 1005       00005         559+           MVI   5(1),14
000670 41E0 0083             00083   560+           LA    14,131(0,0) 000B
000674 40E0 1006             00006   561+           STH   14,6(0,1) STORE LENGTH      000B
000678 41E0 BB04             00B18   562+           LA    14,DD3270
00067C 50E0 1008             00008   563+           ST    14,8(0,1) STORE DCB ADDRESS
000680 5040 100C             0000C   564+           ST    LOOPREG,12(0,1) STORE AREA ADDR   000B
000684 4060 1018             00018   565+           STH   RLNREG,24(0,1) STORE LINE NUMBER
000688 58F0 BB34             00B48   566+           L     15,DD3270+48 LOAD RDWRT ROUT ADDR
00068C 94F7 1004       00004         567+           NI    4(1),X'F7' A38557
000690 05EF                          568+           BALR  14,15 A38557
000692 47F0 B62E             00642   569            B     RSTRKYBD
                                     570 *
000696 5859 200C             0000C   571 SYSOUT     L     INDXREG,12(TABBASE,EVENREG)  GET ADDR OF DATA AREA
00069A D281 B9B4 5000 009C8 00000    572            MVC   SYSOUTBF(130),0(INDXREG)  MOVE DATA TO PRINT BUFFER
0006A0 926B B9D5       009E9         573            MVI   SYSOUTBF+33,C','     MOVE IN
0006A4 926B B9F7       00A0B         574            MVI   SYSOUTBF+67,C','             FIELD DELIMITERS
0006A8 926B BA19       00A2D         575            MVI   SYSOUTBF+101,C','                    FOR LINE
0006AC 9240 BA2F       00A43         576            MVI   SYSOUTBF+123,C' '                            TO BE
0006B0 924B BA35       00A49         577            MVI   SYSOUTBF+129,C'.'                                    PRINTED
0006B4 DC81 B9B4 BE9A 009C8 00EAE    578            TR    SYSOUTBF(130),FLDXLATE    TRANSLATE TO VALIDATE DATA
                                     579            PUT   SYSPRINT,SYSOUTBF    WRITE TO SYSOUT
0006BA 4110 BAB8             00ACC   580+           LA    1,SYSPRINT LOAD PARAMETER REG 1
0006BE 4100 B9B4             009C8   581+           LA    0,SYSOUTBF LOAD PARAMETER REG 0
0006C2 58F0 1030             00030   582+           L     15,48(0,1) LOAD PUT ROUTINE ADDR.
0006C6 05EF                          583+           BALR  14,15 LINK TO PUT ROUTINE
0006C8 47F0 B62E             00642   584            B     RSTRKYBD
                                     585 *
0006CC 957D 8000       00000         586 READFMT3   CLI   0(BUFREG),X'7D'      ENTER KEY
0006D0 4780 B52C             00540   587            BE    WRTRSPNS             YES, WRITE RESPONSE
0006D4 956E 8000       00000         588            CLI   0(BUFREG),X'6E'      PA2 KEY
0006D8 4780 B314             00328   589            BE    ENDMSG               YES, END TERMINAL
0006DC 956C 8000       00000         590            CLI   0(BUFREG),X'6C'      PA1 KEY
0006E0 4770 B412             00426   591            BNE   CNTRLMSG             NO, WRITE CONTROL OPTIONS MSG
                                     592 *
0006E4 5829 300C             0000C   593            L     EVENREG,12(TABBASE,ODDREG)  GET ADDR OF DATA AREA
0006E8 D781 2000 2000 00000 00000    594            XC    0(130,EVENREG),0(EVENREG)   CLEAR IT
0006EE 45E0 B6E8             006FC   595            BAL   RETNREG,WRTFMT1      WRITE OUT FIRST FORMAT
0006F2 12FF                          596            LTR   EPREG,EPREG          WRITE SUCCESSFUL
0006F4 4770 B8EC             00900   597            BNZ   ERRABEND             NO, END
0006F8 47F0 B24C             00260   598            B     READ
                                     599 *
0006FC 183E                          600 WRTFMT1    LR    ODDREG,RETNREG       SAVE RETURN ADDRESS
0006FE 1861                          601            LR    RLNREG,PARMREG       SAVE RLN
000700 D703 7000 7000 00000 00000    602 WRITE1     XC    0(4,DECBREG),0(DECBREG)  CLEAR ECB
                                     603            WRITE (DECBREG),TS,DD3270,FORMAT1,FMT1SZ,,(RLNREG),MF=E
000706 1817                          604+           LR    1,DECBREG LOAD DECB ADDRESS
```

```
  LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                F01OCT71   3/22/72

000708 9404 1004        00004    '   605+           NI    4(1),4
00070C 920E 1005        00005        606+           MVI   5(1),14
000710 47F0 B702              00716  607+           B     *+6 000B
000714 00AE                          608+           DC    AL2(FMT1SZ) 000B
000716 D201 1006 B700 00006 00714    609+           MVC   6(2,1),*-2 MOVE IN LENGTH    000B
00071C 41E0 BB04              00B18  610+           LA    14,DD3270
000720 50E0 1008              00008  611+           ST    14,8(0,1) STORE DCB ADDRESS
000724 41E0 BB4E              00B62  612+           LA    14,FORMAT1 000B
000728 50E0 100C              0000C  613+           ST    14,12(0,1) STORE AREA ADDR   000B
00072C 4060 1018              00018  614+           STH   RLNREG,24(0,1) STORE LINE NUMBER
000730 58F0 BB34              00B48  615+           L     15,DD3270+48 LOAD RDWRT ROUT ADDR
000734 94F7 1004        00004        616+           NI    4(1),X'F7' A38557
000738 05EF                          617+           BALR  14,15 A38557
00073A 12FF                          618            LTR   EPREG,EPREG          END IF RC
00073C 4770 B8EC              00900  619            BNZ   ERRABEND                     NON-ZERO
                                     620            WAIT  ECB=(DECBREG)        WAIT FOR OPERATION TO END
000740 4110 7000              00000  621+           LA    1,0(0,DECBREG) CLEAR HIGH BYTE OF REG
000744 4100 0001              00001  622+           LA    0,1(0,0) COUNT OMITTED,1 USED
000748 0A01                          623+           SVC   1 LINK TO WAIT ROUTINE
00074A 957F 7000        00000        624            CLI   0(DECBREG),X'7F'     SATISFACTORY COMPLETION
00074E 4780 B74C              00760  625            BE    FMT1CD               YES, STORE FORMAT CODE IN ENTRY
000752 9544 7000        00000        626            CLI   0(DECBREG),X'44'     INTERCEPT
000756 4780 B6EC              00700  627            BE    WRITE1               YES, RETRY OPERATION
00075A 41F0 B746              0075A  628            LA    EPREG,*              SET NON-ZERO RETURN CODE
00075E 07F3                          629            BR    ODDREG               AND RETURN TO CALLER
                                     630 *
000760 4120 0001              00001  631 FMT1CD     LA    EVENREG,1            GET FORMAT ID
000764 8B60 0003              00003  632            SLA   RLNREG,3             CONVERT RLN TO TABLE INDEX
000768 4029 6008              00008  633            STH   EVENREG,8(TABBASE,RLNREG)  STORE ID IN TABLE
00076C 8A60 0003              00003  634            SRA   RLNREG,3             RESTORE RLN REGISTER TO RLN
000770 1BFF                          635            SR    EPREG,EPREG          SET RETURN CODE =0
000772 07F3                          636            BR    ODDREG               RETURN TO CALLER
                                     637 *
000774 1B33                          638 PRNTREND   SR    ODDREG,ODDREG
000776 4332 0019              00019  639            IC    ODDREG,RLN(EVENREG)  GET RLN OF ENDING PRINTER
00077A 1863                          640            LR    RLNREG,ODDREG        SAVE RLN
00077C 8B30 0003              00003  641            SLA   ODDREG,3             CONVERT RLN TO TABLE INDEX
000780 5B20 BAB0              00AC4  642            S     EVENREG,FOUR         GET BEGIN OF PRINTER RB
000784 4133 9008              00008  643            LA    ODDREG,8(ODDREG,TABBASE)  GET ADDR OF TABLE ENTRY
000788 5859 0004              00004  644            L     INDXREG,4(TABBASE)   GET ADDR OF ECB LIST
00078C 4A53 0002              00002  645            AH    INDXREG,2(ODDREG)    GET ADDR OF PRINTER'S ECB
000790 957F 2004        00004        646            CLI   4(EVENREG),X'7F'     GOOD END
000794 4780 B790              007A4  647            BE    PRNTRNXT             YES, CLEAN UP
000798 9544 2004        00004        648            CLI   4(EVENREG),X'44'     INTERCEPT
00079C 4780 B8E2              008F6  649            BE    WRITEPI              YES, RESTART OPERATION
0007A0 47F0 B8EC              00900  650            B     ERRABEND             NO, END
                                     651 *
0007A4 D203 3004 2000 00004 00000    652 PRNTRNXT   MVC   4(4,ODDREG),0(EVENREG)   UPDATE LINK POINTER
0007AA 1812                          653            LR    PARMREG,EVENREG      ADDRESS OF PRINTER RB
0007AC 4100 00B8              000B8  654            LA    ZEROREG,184          SIZE OF RB
                                     655            FREEMAIN  R,LV=(0),A=(1)   FREE BLOCK
0007B0 4111 0000              00000  656+           LA    1,0(1) CLEAR THE HIGH ORDER BYTE
0007B4 0A0A                          657+           SVC   10 ISSUE FREEMAIN SVC
0007B6 5823 0004              00004  658            L     EVENREG,4(ODDREG)    GET ADDR OF NEXT RB
0007BA 1222                          659            LTR   EVENREG,EVENREG      END OF CHAIN
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71    3/22/72

0007BC 4770 B7BC            007D0   660            BNZ    WRITEP                NO, START NEXT
0007C0 D703 5000 5000 00000 00000   661            XC     0(4,INDXREG),0(INDXREG)  CLEAR ECB LIST ENTRY
0007C6 D201 3000 BB3C 00000 00B50   662            MVC    0(2,ODDREG),MIN1      MARK PRINTER AVAILABLE
0007CC 47F0 B288            0029C   663            B      TWAIT                 WAIT FOR NEXT COMPLETION
                                    664 *
                                    665 WRITEP     RESETPL  (DECBREG),ATTENT RESET READ INITIAL
0007D0                              666+WRITEP     DS     0H
0007D0 1817                         667+           LR     1,DECBREG
0007D2 1B00                         668+           SR     0,0
0007D4 1BFF                         669+           SR     15,15
0007D6 4301 0019            00019   670+           IC     0,25(1) GET RELATIVE LINE NUMBER FROM DECB.
0007DA 58E1 0008            00008   671+           L      14,8(1) GET DCB ADDRESS FROM DECB.
0007DE 9110 E030      00030         672+           TM     48(14),X'10' HAS DCB BEEN OPENED?
0007E2 4710 B7DA            007EE   673+           BO     IECA0033 IF SO, CONTINUE.
0007E6 41F0 000C            0000C   674+IECJ0033 LA       15,12 IF NOT, SET RETURN CODE   A28622
0007EA 47F0 B880            00894   675+           B      IECB0033 AND EXIT.
0007EE                              676+IECA0033 EQU      *
0007EE 58FE 002C            0002C   677+           L      15,44(14) GET ADDR OF DEB
0007F2 41FF 0000            00000   678+           LA     15,0(15) CLEAR HIGH-ORDER BYTE
0007F6 8B00 0002            00002   679+           SLA    0,2 MULTIPLY RLN BY 4
0007FA 1AF0                         680+           AR     15,0 USE RLN*4 AS INDEX
0007FC 58FF 001C            0001C   681+           L      15,28(15) TO DEBUCBAD-4 AND GET UCB ADDR
000800 9510 F012      00012         682+           CLI    18(15),X'10' DEVICE CLASS = GRAPHICS
000804 4770 B874            00888   683+           BNE    IECO0033 NO, DEVICE IS NOT ANR
000808 9501 1005      00005         684+           CLI    5(1),X'01' IS OPERATION READ INITIAL
00080C 4770 B87E            00892   685+           BNE    IECE0033 NO, GIVE RC=0 AND EXIT
000810 9140 1000      00000         686+           TM     0(1),X'40' IS THE OPERATION COMPLETE
000814 4710 B86C            00880   687+           BO     IECQ0033 YES, GIVE RC=4 AND EXIT
000818 1801                         688+           LR     0,1 DECB ADDR TO REG 0
00081A 181F                         689+           LR     1,15 UCB ADDR TO REG 1
00081C 41F0 0004            00004   690+           LA     15,4 ROUTING CODE OF 4 IN REG 15
000820 0A74                         691+           SVC    116 INVOKE ESR TO DO RESET
000822 4110 0004            00004   692+           LA     1,4 ESTABLISH COMPARAND
000826 19F1                         693+           CR     15,1 IS RC 0 OR 4
000828 4720 B880            00894   694+           BH     IECB0033 NO, EXIT WITH RC
00082C 40F0 B87C            00890   695+           STH    15,IECK0033 SAVE RETURN CODE
                                    696+*
000830 581E 001C            0001C   697+           L      1,28(14) GET IOB BASE FROM DCB
000834 4111 0058            00058   698+           LA     1,88(1) GET ADDR OF FIRST IOB
000838 58EE 002C            0002C   699+           L      14,44(14) GET DEB ADDR FROM DCB
00083C 1B00                         700+           SR     0,0
00083E 430E 0010            00010   701+           IC     0,16(14) GET NO. OF EXTENTS FROM DEB
                                    702+*                         AND USE AS LOOP COUNTER
000842 41EE 0020            00020   703+           LA     14,32(14) GET ADDR OF FIRST DEB UCB PTR
                                    704+*
000846 9601 1001      00001         705+IECT0033 OI       1(1),X'01' SET RESETPL ISSUED FLAG
00084A 9101 101C      0001C         706+           TM     28(1),X'01' RFT IN PROGRESS
00084E 4710 B862            00876   707+           BO     IECW0033 YES, ADJUST RC
000852 58FE 0000            00000   708+           L      15,0(14) GET ADDR OF UCB
000856 9120 F006      00006         709+           TM     6(15),X'20' OUTSTANDING I/O OPERATION
00085A 4710 B84E            00862   710+           BO     IECU0033 YES, KEEP IOB BUSY
00085E 94BF 101C      0001C         711+           NI     28(1),X'BF' TURN OFF IOB BUSY FLAG
                                    712+*
000862 4111 0058            00058   713+IECU0033 LA       1,88(1) STEP TO NEXT IOB
000866 41EE 0004            00004   714+           LA     14,4(14) STEP TO NEXT UCB PTR
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                            F01OCT71   3/22/72

00086A 4600 B832            00846   715+           BCT   0,IECT0033 LOOP IF NOT FINISHED
00086E 48F0 B87C            00890   716+           LH    15,IECK0033 RESTORE RETURN CODE
000872 47F0 B880            00894   717+           B     IECB0033 EXIT WITH RC
                                    718+*
000876 D701 B87C B87C 00890 00890   719+IECW0033 XC     IECK0033,IECK0033 SET RC TO ZERO
00087C 47F0 B84E            00862   720+           B     IECU0033
                                    721+*
000880 41F0 0004            00004   722+IECQ0033 LA     15,4 SET RETURN CODE OF 4
000884 47F0 B880            00894   723+           B     IECB0033 AND EXIT
                                    724+*
000888 41F0 0008            00008   725+IECO0033 LA     15,8 INVALID DEVICE, SET RC = 8
00088C 47F0 B880            00894   726+           B     IECB0033 AND EXIT
000890 0000                         727+IECK0033 DC     H'0' SAVE AREA FOR WTTA                                   0003
000892 1BFF                         728+IECE0033 SR     15,15 000A
000894                              729+IECB0033 EQU    *
000894 12FF                         730            LTR   EPREG,EPREG         RC = 0
000896 4780 B898            008AC   731            BZ    WRITEPS             YES, PROCEED
00089A 59F0 BAB0            00AC4   732            C     EPREG,FOUR          RC = 4
00089E 4770 B8EC            00900   733            BNE   ERRABEND            NO, END
                                    734            WAIT  ECB=(DECBREG)       WAIT FOR READ TO COMPLETE
0008A2 4110 7000            00000   735+           LA    1,0(0,DECBREG) CLEAR HIGH BYTE OF REG
0008A6 4100 0001            00001   736+           LA    0,1(0,0) COUNT OMITTED,1 USED
0008AA 0A01                         737+           SVC   1 LINK TO WAIT ROUTINE
                                    738 *
0008AC 4122 0004            00004   739 WRITEPS    LA    EVENREG,4(EVENREG)  GET ADDR OF ECB
0008B0 5025 0000            00000   740            ST    EVENREG,0(INDXREG)  STORE ADDR OF ECB IN ECB LIST
0008B4 4132 0028            00028   741            LA    ODDREG,40(EVENREG)  GET ADDR OF PRINT BUFFER
                                    742            WRITE (EVENREG),TS,DD3270,(ODDREG),131,,(RLNREG),MF=E
0008B8 1812                         743+           LR    1,EVENREG LOAD DECB ADDRESS
0008BA 9404 1004      00004         744+           NI    4(1),4
0008BE 920E 1005      00005         745+           MVI   5(1),14
0008C2 41E0 0083            00083   746+           LA    14,131(0,0) 000B
0008C6 40E0 1006            00006   747+           STH   14,6(0,1) STORE LENGTH       000B
0008CA 41E0 BB04            00B18   748+           LA    14,DD3270
0008CE 50E0 1008            00008   749+           ST    14,8(0,1) STORE DCB ADDRESS
0008D2 5030 100C            0000C   750+           ST    ODDREG,12(0,1) STORE AREA ADDR   000B
0008D6 4060 1018            00018   751+           STH   RLNREG,24(0,1) STORE LINE NUMBER
0008DA 58F0 BB34            00B48   752+           L     15,DD3270+48 LOAD RDWRT ROUT ADDR
0008DE 94F7 1004      00004         753+           NI    4(1),X'F7' A38557
0008E2 05EF                         754+           BALR  14,15 A38557
0008E4 12FF                         755            LTR   EPREG,EPREG         END IF RC
0008E6 4770 B8EC            00900   756            BNZ   ERRABEND                NON-ZERO
0008EA 9548 7000      00000         757            CLI   0(DECBREG),X'48'    READ INITIAL ENDED BY RESETPL
0008EE 4780 B24C            00260   758            BE    READ                YES, REISSUE READ
0008F2 47F0 B288            0029C   759            B     TWAIT               NO, CHECK STATUS
                                    760 *
0008F6 D703 2004 2004 00004 00004   761 WRITEPI    XC    4(4,EVENREG),4(EVENREG)  CLEAR ECB
0008FC 47F0 B7BC            007D0   762            B     WRITEP              RESTART WRITE
                                    763 *
                                    764 *
                                    765 ERRABEND   ABEND (15),DUMP
000900                              766+ERRABEND DS     0H
000900 181F                         767+           LR    1,15 LOAD PARAMETER REG 1
000902 4100 0080            00080   768+           LA    0,128 PICK UP DUMP/STEP CODE
000906 8900 0018            00018   769+           SLL   0,24(0) SHIFT TO HIGH BYTE
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                F01OCT71    3/22/72

00090A 1610                         770+          OR    1,0 OR IN WITH COMPCODE
00090C 0A0D                         771+          SVC   13 LINK TO ABEND ROUTINE
00090E 1B22                         772 GETPARM   SR    EVENREG,EVENREG       CLEAR REGISTER
000910 1B66                         773           SR    CNTREG,CNTREG
000912 95F0 5000          00000     774 GETCHAR   CLI   0(INDXREG),C'0'       CHARACTER A NUMBER
000916 4740 B94C                00960  775           BL    DELIM                 NO, CHECK FOR DELIMITER
00091A 95F9 5000          00000     776           CLI   0(INDXREG),C'9'       MAYBE
00091E 4720 B968                0097C  777           BH    INVLDCHR              NO, INVALID CHARACTER
000922 8920 0008                00008  778           SLL   EVENREG,8             CLEAR LOW CHARACTER
000926 4325 0000                00000  779           IC    EVENREG,0(INDXREG)    GET CHARACTER
00092A 4166 0001                00001  780 PARMLOOP  LA    CNTREG,1(CNTREG)      INCR LENGTH
00092E 4155 0001                00001  781           LA    INDXREG,1(INDXREG)    INCR STRING POINTER
000932 4640 B8FE                00912  782           BCT   LOOPREG,GETCHAR       GET NEXT CHARACTER, IF ANY LEFT
000936 5960 BAA8                00ABC  783 HAVEPARM  C     CNTREG,THREE          CHECK PARM LENGTH
00093A 47D0 B932                00946  784           BNH   CNVRT                 BR IF VALID
00093E 5420 BAAC                00AC0  785           N     EVENREG,SIXFS         REDUCE TO THREE BYTES
000942 5860 BAA8                00ABC  786           L     CNTREG,THREE          SET COUNT TO THREE
000946 5020 BAA4                00AB8  787 CNVRT     ST    EVENREG,RAWPARM       STORE RAW PARM VALUE
00094A 4130 0004                00004  788           LA    ODDREG,4              LENGTH OF RAWPARM
00094E 1B36                         789           SR    ODDREG,CNTREG         SUBT LENGTH OF PARM &
000950 4133 BAA4                00AB8  790           LA    ODDREG,RAWPARM(ODDREG) GET ADDR OF FIRST CHARACTER
000954 0660                         791           BCTR  CNTREG,0              DECR COUNT FOR EXECUTE
000956 4460 B974                00988  792           EX    CNTREG,PACK           PACK PARM
00095A 4F10 B9AC                009C0  793           CVB   PARMREG,PCKDPARM      & CONVERT TO BINARY
00095E 07FE                         794           BR    RETNREG               RETURN TO CALLER
                                    795 *
000960 956B 5000          00000     796 DELIM     CLI   0(INDXREG),C','       COMMA
000964 4770 B968                0097C  797           BNE   INVLDCHR              NO, TREAT AS INVALID
000968 1266                         798           LTR   CNTREG,CNTREG         PARM OMITTED
00096A 4780 B964                00978  799           BZ    ZEROPARM              YES, SET IT TO ZERO
00096E 4155 0001                00001  800           LA    INDXREG,1(INDXREG)    INCR STRING POINTER
000972 0640                         801           BCTR  LOOPREG,0             DECR LOOP COUNTER
000974 47F0 B922                00936  802           B     HAVEPARM              BR TO CONVERT PARM
                                    803 *
000978 1B11                         804 ZEROPARM  SR    PARMREG,PARMREG       SET PARM = 0
00097A 07FE                         805           BR    RETNREG               RETURN
00097C 8920 0008                00008  806 INVLDCHR  SLL   EVENREG,8             CLEAR LOW CHARACTER
000980 4320 BB3E                00B52  807           IC    EVENREG,C0            SUBSTITUTE C'0' FOR INVALID CHAR
000984 47F0 B916                0092A  808           B     PARMLOOP              BR TO ADJUST COUNTERS & POINTERS
                                    809 *
000988 F270 B9AC 3000 009C0 00000   810 PACK      PACK  PCKDPARM(8),0(0,ODDREG)
                                    811 *
00098E 4125 0003                00003  812 GETFIELD  LA    EVENREG,3(INDXREG)    INCR TO FIRST DATA BYTE OF FIELD
000992 5B40 BAA8                00ABC  813           S     LOOPREG,THREE         CORRECT LENGTH FOR SBA SEQUENCE
000996 1B66                         814           SR    CNTREG,CNTREG         SET COUNT = ZERO
000998 9511 2000          00000     815 FNDLOOP   CLI   0(EVENREG),X'11'      BYTE = SBA ORDER
00099C 4780 B998                009AC  816           BE    MVFLD                 YES, MOVE FIELD
0009A0 4166 0001                00001  817           LA    CNTREG,1(CNTREG)      INCR FIELD SIZE
0009A4 4122 0001                00001  818           LA    EVENREG,1(EVENREG)    INCR TO NEXT DATA BYTE
0009A8 4640 B984                00998  819           BCT   LOOPREG,FNDLOOP       LOOP IF MORE DATA
0009AC 0660                         820 MVFLD     BCTR  CNTREG,0              ADJUST COUNT FOR EXECUTE
0009AE 4460 B9A2                009B6  821           EX    CNTREG,FLDMOVE        MOVE FIELD INTO DATA AREA
0009B2 1852                         822           LR    INDXREG,EVENREG       GET START OF NEXT FIELD
0009B4 07FE                         823           BR    RETNREG               RETURN
                                    824 *
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                F01OCT71    3/22/72

0009B6 D200 3000 5003 00000 00003   825 FLDMOVE  MVC    0(0,ODDREG),3(INDXREG) MOVE FIELD FROM BUFFER
                                    826 *                                TO DATA AREA
0009C0                              827          DS     0D
0009C0 F0F0F0F0F0F0F0F0             828 PCKDPARM DC     8C'0'                PACKED PARM AREA
0009C8                              829 SYSOUTBF DS     0D                   SYSOUT PRINT BUFFER
0009C8                              830          DS     CL130
000A4C                              831 SAVE     DS     18F                  SAVE AREA
000A94 000000FF                     832 MAXRLN   DC     F'255'               MAXIMUM RLN
000A98 00000001                     833 NODVCS   DC     F'1'                 NO. OF DEVICES (DEFAULT = 1)
000A9C 00000000                     834 NOPTRS   DC     F'0'                 NO. OF PRINTERS (DEFAULT = 0)
000AA0                              835 TABSIZE  DS     F                    SIZE OF CONTROL TABLE
000AA4 00000100                     836 F256     DC     F'256'
000AA8 00000082                     837 DATASZ   DC     F'130'               SIZE OF DISPLAY TERM DATA AREA
000AAC                              838 DATBLKSZ DS     F                    SIZE OF DATA AREA CORE BLOCK
000AB0                              839 DATBLKAD DS     F                    ADDR OF DATA AREA CORE BLOCK
000AB4 00000001                     840 READRLN  DC     F'1'                 RLN TO BE USED FOR READ INITIALS
000AB8                              841 RAWPARM  DS     1F                   ZONED PARM AREA
000ABC 00000003                     842 THREE    DC     F'3'
000AC0 00FFFFFF                     843 SIXFS    DC     XL4'00FFFFFF'
000AC4 00000004                     844 FOUR     DC     F'4'
000AC8 00000000                     845 ZEROWRD  DC     F'0'
                                    846 SYSPRINT DCB    DDNAME=SYSPRINT,DSORG=PS,DEVD=DA,MACRF=(PM),RECFM=F,LREC*
                                                        L=130,BLKSIZE=130

                                    848+*                          DATA CONTROL BLOCK
                                    849+*
000ACC                              850+SYSPRINT DC     0F'0' ORIGIN ON WORD BOUNDARY

                                    852+*                          DIRECT ACCESS DEVICE INTERFACE

000ACC 0000000000000000             854+         DC     BL16'0' FDAD,DVTBL
000ADC 00000000                     855+         DC     A(0) KEYLE,DEVT,TRBAL

                                    857+*                          COMMON ACCESS METHOD INTERFACE

000AE0 00                           859+         DC     AL1(0) BUFNO
000AE1 000001                       860+         DC     AL3(1) BUFCB
000AE4 0000                         861+         DC     AL2(0) BUFL
000AE6 4000                         862+         DC     BL2'0100000000000000' DSORG
000AE8 00000001                     863+         DC     A(1) IOBAD

                                    865+*                          FOUNDATION EXTENSION

000AEC 00                           867+         DC     BL1'00000000' BFTEK,BFLN,HIARCHY
000AED 000001                       868+         DC     AL3(1) EODAD
000AF0 80                           869+         DC     BL1'10000000' RECFM
000AF1 000000                       870+         DC     AL3(0) EXLST

                                    872+*                          FOUNDATION BLOCK

000AF4 E2E8E2D7D9C9D5E3             874+         DC     CL8'SYSPRINT' DDNAME
000AFC 02                           875+         DC     BL1'00000010' OFLGS
000AFD 00                           876+         DC     BL1'00000000' IFLG
000AFE 0050                         877+         DC     BL2'0000000001010000' MACR
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                      F01OCT71   3/22/72

                                   879+*                        BSAM-BPAM-QSAM INTERFACE

000B00 00                          881+          DC    BL1'00000000' RER1
000B01 000001                      882+          DC    AL3(1) CHECK, GERR, PERR
000B04 00000001                    883+          DC    A(1) SYNAD
000B08 0000                        884+          DC    H'0' CIND1, CIND2
000B0A 0082                        885+          DC    AL2(130) BLKSIZE
000B0C 00000000                    886+          DC    F'0' WCPO, WCPL, OFFSR, OFFSW
000B10 00000001                    887+          DC    A(1) IOBA
000B14 00                          888+          DC    AL1(0) NCP
000B15 000001                      889+          DC    AL3(1) EOBR, EOBAD

                                   891+*                             QSAM INTERFACE

000B18 00000001                    893+          DC    A(1) RECAD
000B1C 0000                        894+          DC    H'0' QSWS
000B1E 0082                        895+          DC    AL2(130) LRECL
000B20 00                          896+          DC    BL1'00000000' EROPT
000B21 000001                      897+          DC    AL3(1) CNTRL
000B24 00000000                    898+          DC    F'0' PRECL
000B28 00000001                    899+          DC    A(1) EOB
                                   900 DD3270    DCB   DDNAME=DD3270,DSORG=CX,MACRF=(R,W),EROPT=T

                                   902+*                        DATA CONTROL BLOCK
                                   903+*
000B18                             904+          ORG   *-20 TO ELIMINATE UNUSED SPACE
000B18                             905+DD3270    DS    0F'0' ORIGIN ON WORD BOUNDARY
000B2C                             906+          ORG   *+20 TO ORIGIN GENERATION

                                   908+*                        COMMON ACCESS METHOD INTERFACE

000B2C 00                          910+          DC    AL1(0) BUFNO
000B2D 000001                      911+          DC    AL3(1) BUFCB
000B30 0000                        912+          DC    AL2(0) BUFL
000B32 1000                        913+          DC    BL2'0001000000000000' DSORG
000B34 00000001                    914+          DC    A(1) IOBAD

                                   916+*                        FOUNDATION EXTENSION

000B38 00                          918+          DC    BL1'00000000' BFTEK,BFLN,HIARCHY
000B39 10                          919+          DC    BL1'00010000' BTAM EROPT CODE
000B3A FF                          920+          DC    AL1(255) BTAM BUFFER COUNT
000B3B 00                          921+          DC    AL1(0)
000B3C 00                          922+          DC    BL1'00000000' RECFM
000B3D 000000                      923+          DC    AL3(0) EXLST

                                   925+*                        FOUNDATION BLOCK

000B40 C4C4F3F2F7F04040            927+          DC    CL8'DD3270' DDNAME
000B48 02                          928+          DC    BL1'00000010' OFLGS
000B49 00                          929+          DC    BL1'00000000' IFLG
000B4A 2020                        930+          DC    BL2'0010000000100000' MACR

                                   932+*                        BTAM INTERFACE
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                   F01OCT71   3/22/72

000B4C 00000000                     934+            DC    F'0'
000B50 FFFF                         935 MIN1        DC    H'-1'             FREE PRINTER FORMAT ID
000B52 F0                           936 C0          DC    C'0'
000B53 11C1D6                       937 FLD1        DC    XL3'11C1D6'       SBA SEQ OF FIRST FIELD ( SBA = 86)
000B56 11C17E                       938 FLD2        DC    XL3'11C17E'       SBA SEQ OF SECOND FIELD( SBA =126)
000B59 11C2E6                       939 FLD3        DC    XL3'11C2E6'       SBA SEQ OF THIRD FIELD ( SBA =166)
000B5C 11C34F                       940 FLD4        DC    XL3'11C34F'       SBA SEQ OF FOURTH FIELD (SBA =207)
000B5F 11C36A                       941 FLD5        DC    XL3'11C36A'       SBA SEQ OF FIFTH FIELD ( SBA =234)
000B62 C7114040                     942 FORMAT1     DC    X'C7114040'        WCC, SBA = 0
000B66 C5D5E3C5D940C4C1             943             DC    C'ENTER DATA REQUESTED BELOW:'
000B81 11C150                       944             DC    X'11C150'         SBA = 80
000B84 D5C1D4C57A                   945             DC    C'NAME:'
000B89 1D401311C1F71D60             946             DC    X'1D401311C1F71D60' SF = UNPROT, IC, SBA=119, SF=PROT
000B91 C1C4C4D97A                   947             DC    C'ADDR:'
000B96 1D4011C25F1D60               948             DC    X'1D4011C25F1D60'   SF = UNPROT, SBA = 159, SF = PROT
000B9D C3C9E3E87A                   949             DC    C'CITY:'
000BA2 1D4011C3C71D60               950             DC    X'1D4011C3C71D60'   SF = UNPROT, SBA=199, SF=PROT
000BA9 E2E3C1E3C57A                 951             DC    C'STATE:'
000BAF 1D4011C3E41D60               952             DC    X'1D4011C3E41D60'   SF = UNPROT, SBA=228, SF=PROT
000BB6 E9C9D77A                     953             DC    C'ZIP:'
000BBA 1D5011C36F1D6011             954             DC    X'1D5011C36F1D6011C540'  SF=UNPROT, SBA=239, SF=PROT,
                                    955 *                                   SBA= 320
000BC4 C5D5E3C5D940D2C5             956             DC    C'ENTER KEY: ENTER DATA;'
000BDA 11C5E8                       957             DC    X'11C5E8'         SBA = 360
000BDD D7C1F240D2C5E87A             958             DC    C'PA2 KEY: END PROGRAM;'
000BF2 11C650                       959             DC    X'11C650'         SBA = 400
000BF5 C3D3C5C1D940D2C5             960             DC    C'CLEAR KEY: CONTROL OPTIONS.'
0000AE                              961 FMT1SZ      EQU   *-FORMAT1
                                    962 *
000C10 C7114040                     963 FORMAT2     DC    X'C7114040'       WCC, SBA = 0
000C14 C4C1E3C140C7C9E5             964             DC    C'DATA GIVEN BELOW ENTERED:'
000C2D 11C150                       965             DC    X'11C150'         SBA = 80
000C30 D5C1D4C57A                   966             DC    C'NAME:'
000C35 1D4013                       967             DC    X'1D4013'         SF=UNPROT, IC
000028                              968 FMT2FLD1    EQU   *-FORMAT2
000C38 0000000000000000             969             DC    XL33'0'           DATA FIELD 1
000C59 1D60                         970             DC    X'1D60'           SF = PROT
000C5B C1C4C4D97A                   971             DC    C'ADDR:'
000C60 1D40                         972             DC    X'1D40'           SF= UNPROT
000052                              973 FMT2FLD2    EQU   *-FORMAT2
000C62 0000000000000000             974             DC    XL33'0'           DATA FIELD 2
000C83 1D60                         975             DC    X'1D60'           SF = PROT
000C85 C3C9E3E87A                   976             DC    C'CITY:'
000C8A 1D40                         977             DC    X'1D40'           SF = UNPROT
00007C                              978 FMT2FLD3    EQU   *-FORMAT2
000C8C 0000000000000000             979             DC    XL33'0'           DATA FIELD 3
000CAD 1D60                         980             DC    X'1D60'           SF = PROT
000CAF E2E3C1E3C57A                 981             DC    C'STATE:'
000CB5 1D40                         982             DC    X'1D40'           SF = UNPROT
0000A7                              983 FMT2FLD4    EQU   *-FORMAT2
000CB7 0000000000000000             984             DC    XL21'0'
000CCC 1D60                         985             DC    X'1D60'           SF = PROT
000CCE E9C9D77A                     986             DC    C'ZIP:'
000CD2 1D50                         987             DC    X'1D50'           SF = UNPROT, NUM ONLY
0000C4                              988 FMT2FLD5    EQU   *-FORMAT2
```

```
  LOC  OBJECT CODE      ADDR1 ADDR2  STMT    SOURCE STATEMENT                                                  F01OCT71    3/22/72

000CD4 0000000000                     989           DC    XL5'0'
000CD9 1D6011C540                     990           DC    X'1D6011C540'        SF = PROT, SBA = 320
000CDE C5D5E3C5D940D2C5               991           DC    C'ENTER KEY: UPDATE DATA;'
000CF5 11C5E8                         992           DC    X'11C5E8'            SBA = 360
000CF8 D7C1F140D2C5E87A               993           DC    C'PA1 KEY: PRINT DATA;'
000D0C 11C650                         994           DC    X'11C650'            SBA = 400
000D0F D7C1F240D2C5E87A               995           DC    C'PA2 KEY: END PROGRAM;'
000D24 11C6F8                         996           DC    X'11C6F8'            SBA = 440
000D27 C3D3C5C1D940D2C5               997           DC    C'CLEAR KEY: CONTROL OPTIONS.'
000132                                998 FMT2SZ    EQU   *-FORMAT2
                                      999 *
000D42 C711404013                    1000 FORMAT3   DC    X'C711404013'        WCC, SBA =0, IC
000D47 E7E7E8E8F3C3E4E4              1001           DC    C'XXYY3CUU'
000D4F 11C150                        1002           DC    X'11C150'            SBA = 80
000D52 C5D5E3C5D940D2C5              1003           DC    C'ENTER KEY: RESUME AND CONTINUE;'
000D71 11C1F8                        1004           DC    X'11C1F8'            SBA = 120
000D74 D7C1F140D2C5E87A              1005           DC    C'PA1 KEY: BEGIN NEW ENTRY;'
000D8D 11C260                        1006           DC    X'11C260'            SBA = 160
000D90 D7C1F240D2C5E87A              1007           DC    C'PA2 KEY: END PROGRAM;'
000DA5 11C3F0                        1008           DC    X'11C3F0'            SBA = 240
                                     1009           DC    C'TO REQUEST BTAM OLT -- ENTER REQUEST FOR TEST MESSAGE *
000DA8 E3D640D9C5D8E4C5                                   OVER SAMPLE FORMAT ABOVE:  '
000DF9 E7E77EE3C5E2E340              1010           DC    C'XX=TEST NO. (23-28)  YY=REPEATS (01-99)'
000E20 40C3E4E47EC1C4C4              1011           DC    C' CUU=ADDRESS OF TARGET DEVICE'
000E3D 11C6D1                        1012           DC    X'11C6D1'            SBA = 401
                                     1013           DC    C'THEN HIT ERASE EOF AND THEN TEST REQ.  USE CLEAR KEY T*
000E40 E3C8C5D540C8C9E3                                   O RESUME AFTER TEST.'
000148                               1014 FMT3SZ    EQU   *-FORMAT3
                                     1015 *
000E8A C711C17D                      1016 FORMAT4   DC    X'C711C17D'          WCC, SBA = 125
000E8E D3D6C3C1D340F3F2              1017           DC    C'LOCAL 3270 SAMPLE PROGRAM ENDED.'
000024                               1018 FMT4SZ    EQU   *-FORMAT4
                                     1019 *
                                     1020 *
                                     1021 *             0123456789ABCDEF
000EAE 4040404040404040              1022 FLDXLATE DC    C'                ' 0
000EBE 4040404040404040              1023           DC    C'                ' 1
000ECE 4040404040404040              1024           DC    C'                ' 2
000EDE 4040404040404040              1025           DC    C'                ' 3
000EEE 4040404040404040              1026           DC    C'          ←.<(+|' 4
000EFE 5040404040404040              1027           DC    C'&&         !$*); ' 5
000F0E 6061404040404040              1028           DC    C'-/        ,%_>?' 6
000F1E 4040404040404040              1029           DC    C'          :#@''="' 7
000F2E 40C1C2C3C4C5C6C7              1030           DC    C' ABCDEFGHI      ' 8
000F3E 40D1D2D3D4D5D6D7              1031           DC    C' JKLMNOPQR      ' 9
000F4E 4040E2E3E4E5E6E7              1032           DC    C'  STUVWXYZ      ' A
000F5E 4040404040404040              1033           DC    C'                ' B
000F6E 40C1C2C3C4C5C6C7              1034           DC    C' ABCDEFGHI      ' C
000F7E 40D1D2D3D4D5D6D7              1035           DC    C' JKLMNOPQR      ' D
000F8E 4040E2E3E4E5E6E7              1036           DC    C'  STUVWXYZ      ' E
000F9E F0F1F2F3F4F5F6F7              1037           DC    C'0123456789      ' F
                                     1038 *             0123456789ABCDEF
                                     1039           END
```

3/22/72

POS.ID	REL.ID	FLAGS	ADDRESS
01	01	08	0001B1
01	01	08	0001B5
01	01	08	0003E5
01	01	08	0003E9

3/22/72

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BASEREG	00001	00000B	00042	0057 0058 0059
BASEREG2	00001	00000A	00041	0058 0059 0060 0060
BUFREG	00001	000008	00039	0205 0247 0298 0389 0426 0429 0477 0478 0483 0484 0485 0486 0487 0498 0515
				0586 0588 0590
CLRLOOP	00004	0000B8	00113	0118
CLRMOVE	00006	0000E0	00125	0122
CLR1	00004	00041E	00389	0299
CNTREG	00001	000006	00034	0035 0166 0166 0172 0185 0226 0231 0231 0234 0235 0288 0289 0290 0773 0773
				0780 0780 0783 0786 0789 0791 0792 0798 0798 0814 0814 0817 0817 0820 0821
CNTRLMSG	00002	000426	00393	0591
CNVRT	00004	000946	00787	0784
COUNT	00001	000012	00048	0431
CO	00001	000B52	00936	0807
DATASZ	00004	000AA8	00837	0157 0174
DATBLKAD	00004	000AB0	00839	0163 0369
DATBLKSZ	00004	000AAC	00838	0159 0370
DD3270	00004	000B18	00905	0193 0245 0249 0308 0338 0343 0367 0406 0411 0496 0500 0562 0566 0610 0615
				0748 0752
DECBREG	00001	000007	00037	0204 0238 0238 0240 0330 0330 0332 0349 0352 0354 0398 0398 0400 0417 0420
				0422 0488 0488 0490 0506 0509 0511 0602 0602 0604 0621 0624 0626 0667 0735
				0757
DEFAULTS	00004	000068	00089	0069
DELIM	00004	000960	00796	0775
DFLTNO	00002	000052	00079	0073 0075
DFLTPTR	00004	000068	00090	0080 0084
DFLTRLN	00002	000128	00148	0133
DSPLY1	00004	00025C	00235	0230
ENDLIST	00002	0000F2	00132	0136 0138
ENDMSG	00002	000328	00302	0589
ENT1	00004	000492	00426	0390
EPREG	00001	00000F	00045	0209 0213 0214 0214 0252 0252 0346 0346 0414 0414 0503 0503 0596 0596 0618
				0618 0628 0635 0635 0730 0730 0732 0755 0755
ERRABEND	00002	000900	00766	0253 0284 0291 0347 0356 0415 0424 0504 0513 0597 0619 0650 0733 0756
EVENREG	00001	000002	00030	0062 0063 0064 0087 0092 0095 0103 0110 0112 0115 0115 0116 0116 0125 0125
				0219 0220 0220 0221 0274 0279 0281 0283 0285 0304 0305 0359 0360 0385 0396
				0397 0431 0480 0481 0517 0518 0519 0530 0545 0571 0593 0594 0594 0631 0633
				0639 0642 0646 0648 0652 0653 0658 0659 0659 0739 0739 0740 0741 0743 0761
				0761 0772 0772 0778 0779 0785 0787 0806 0807 0812 0815 0818 0818 0822
FINDFRST	00006	000242	00229	0233
FININIT	00004	000136	00154	0131 0149
FLDMOVE	00006	0009B6	00825	0821
FLDXLATE	00016	000EAE	01022	0578
FLD1	00003	000B53	00937	0436
FLD2	00003	000B56	00938	0444
FLD2CHK	00004	0004CE	00443	0437
FLD3	00003	000B59	00939	0452
FLD3CHK	00004	0004EC	00451	0445
FLD4	00003	000B5C	00940	0460
FLD4CHK	00004	00050A	00459	0453
FLD5	00003	000B5F	00941	0468
FLD5CHK	00004	000528	00467	0461
FMT1CD	00004	000760	00631	0625
FMT1SZ	00001	0000AE	00961	0608
FMT2FLD1	00001	000028	00968	0483
FMT2FLD2	00001	000052	00973	0484

```
SYMBOL    LEN   VALUE  DEFN     REFERENCES                                                                           3/22/72

FMT2FLD3 00001 00007C 00978   0485
FMT2FLD4 00001 0000A7 00983   0486
FMT2FLD5 00001 0000C4 00988   0487
FMT2SZ   00001 000132 00998   0478  0494
FMT3SZ   00001 000148 01014   0404
FMT4SZ   00001 000024 01018   0336
FNDLOOP  00004 000998 00815   0819
FORMAT1  00004 000B62 00942   0612  0961
FORMAT2  00004 000C10 00963   0477  0478  0968  0973  0978  0983  0988  0998
FORMAT3  00005 000D42 01000   0408  1014
FORMAT4  00004 000E8A 01016   0340  1018
FOUR     00004 000AC4 00844   0642  0732
F256     00004 000AA4 00836   0113  0117
GETCHAR  00004 000912 00774   0782
GETFIELD 00004 00098E 00812   0439  0447  0455  0463  0471
GETPARM  00002 00090E 00772   0071  0082  0134
GETPTRS  00002 0000E6 00128   0119  0123
GETTAB   00004 000068 00087   0089  0090
HAVEPARM 00004 000936 00783   0802
IECA0008 00001 0002A0 00257   0273
IECA0010 00004 000376 00327   0314  0320
IECA0033 00001 0007EE 00676   0673
IECB0008 00004 0002A6 00260   0266
IECB0010 00001 00037A 00328   0326
IECB0033 00001 000894 00729   0675  0694  0717  0723  0726
IECC0008 00004 0002D6 00274   0262
IECD0008 00002 0002C2 00267   0264
IECE0033 00002 000892 00728   0685
IECJ0033 00004 0007E6 00674
IECK0033 00002 000890 00727   0695  0716  0719  0719
IECO0033 00004 000888 00725   0683
IECQ0033 00004 000880 00722   0687
IECT0033 00004 000846 00705   0715
IECU0033 00004 000862 00713   0710  0720
IECW0033 00006 000876 00719   0707
IHB0005A 00002 0001E6 00201   0197
INCRPTRS 00004 00017C 00175   0186
INDXREG  00001 000005 00033   0070  0164  0171  0171  0173  0174  0179  0180  0181  0206  0210  0216  0216  0223  0223
                              0227  0229  0232  0232  0429  0436  0444  0452  0460  0468  0530  0532  0542  0543  0551
                              0552  0553  0571  0572  0644  0645  0661  0661  0740  0774  0776  0779  0781  0781  0796
                              0800  0800  0812  0822  0825
INDXREG2 00001 000008 00038   0129  0129  0142  0143  0143  0165  0169  0172  0173  0175  0175
INITLOOP 00006 000160 00169   0177
INITPTR  00004 00010A 00139   0151
INVLDCHR 00004 00097C 00806   0777  0797
IOINIT   00004 0001AC 00189   0184
IOLOOP   00004 0001FC 00209   0224
IOLPCNT  00004 00022A 00222   0217
LASTMOVE 00002 0000D6 00121   0114
LOOPREG  00001 000004 00032   0067  0068  0068  0078  0078  0132  0132  0156  0177  0203  0224  0228  0233  0430  0431
                              0432  0433  0433  0440  0440  0448  0448  0456  0456  0464  0464  0529  0531  0532  0533
                              0534  0535  0536  0537  0564  0782  0801  0813  0819
LOOPREG2 00001 000007 00036   0128  0128  0130  0144  0144  0148  0148  0167  0176  0176  0185
MAXRLN   00004 000A94 00832   0074
MIN1     00002 000B50 00935   0141  0169  0210  0229  0540  0662
```

```
SYMBOL    LEN   VALUE  DEFN     REFERENCES                                                                              3/22/72

MVFLD     00002 0009AC 00820    0816
NODVCS    00004 000A98 00833    0076  0083  0087  0101  0137  0154  0203  0228
NOPTRS    00004 000A9C 00834    0085  0093  0130  0155  0179
ODDREG    00001 000003 00031    0066  0067  0070  0093  0094  0095  0105  0110  0254  0258  0286  0287  0288  0428  0428
                                0438  0438  0443  0443  0446  0446  0451  0451  0454  0454  0459  0459  0462  0462  0467
                                0467  0470  0470  0482  0483  0484  0485  0486  0487  0519  0520  0520  0522  0523  0539
                                0542  0544  0549  0552  0555  0593  0600  0629  0636  0638  0638  0639  0640  0641  0643
                                0643  0645  0652  0658  0662  0741  0750  0788  0789  0790  0790  0810  0825
PACK      00006 000988 00810    0792
PARMLOOP  00004 00092A 00780    0808
PARMREG   00001 000001 00029    0066  0072  0072  0074  0076  0083  0085  0100  0103  0104  0105  0105  0106  0107  0107
                                0108  0110  0113  0117  0121  0122  0135  0135  0137  0139  0140  0140  0141  0142  0150
                                0154  0155  0156  0158  0163  0164  0178  0181  0181  0182  0183  0212  0285  0286  0301
                                0303  0305  0369  0374  0392  0395  0397  0474  0479  0481  0482  0517  0528  0528  0529
                                0543  0544  0545  0546  0549  0550  0550  0553  0557  0601  0653  0793  0804  0804
PARM2     00002 000052 00078    0079
PCKDPARM  00001 0009C0 00828    0793  0810
POLPT     00001 000024 00050    0285
PRNTREND  00002 000774 00638    0280
PRNTRNXT  00006 0007A4 00652    0647
PTRINIT   00002 0001A6 00185    0170
PTRLOOP   00004 0000EA 00130    0145
QCNTSTR   00004 000416 00385    0360
QUIESCE   00004 0003D8 00359    0353
RAWPARM   00004 000AB8 00841    0787  0790
READ      00004 000260 00237    0282  0295  0386  0421  0427  0510  0598  0758
READFMT1  00004 000320 00298    0292  0516
READFMT2  00004 0005D6 00515    0293
READFMT3  00004 0006CC 00586    0294
READRLN   00004 000AB4 00840    0235  0237
READRTN   00004 00030C 00291    0290
RETNREG   00001 00000E 00044    0071  0082  0134  0213  0439  0447  0455  0463  0471  0595  0600  0794  0805  0823
RLN       00001 000019 00049    0639
RLNREG    00001 000006 00035    0207  0212  0222  0222  0237  0248  0301  0316  0342  0392  0410  0474  0499  0522  0565
                                0601  0614  0632  0633  0634  0640  0751
RSTRKYBD  00002 000642 00545    0569  0584
SAMP327L  00001 000000 00002
SAVE      00004 000A4C 00831    0061  0062  0379
SAVEREG   00001 00000D 00043    0061  0063  0064  0379
SIXFS     00004 000AC0 00843    0785
STRTPRTR  00004 00064C 00549    0541
SYSOUT    00004 000696 00571    0521
SYSOUTBF  00008 0009C8 00829    0572  0573  0574  0575  0576  0577  0578  0581
SYSPRINT  00004 000ACC 00850    0191  0365  0580
TABBASE   00001 000009 00040    0100  0101  0102  0102  0103  0104  0106  0108  0111  0112  0140  0165  0178  0183  0204
                                0205  0206  0219  0221  0227  0254  0288  0305  0359  0374  0385  0397  0428  0481  0482
                                0519  0530  0539  0542  0544  0549  0551  0552  0555  0571  0593  0633  0643  0644
TABSIZE   00004 000AA0 00835    0096  0375
THREE     00004 000ABC 00842    0432  0783  0786  0813
TWAIT     00004 00029C 00254    0663  0759
TYPE      00001 000005 00047    0279
UPCOUNT   00004 00021E 00219    0211  0215
WRITEP    00002 0007D0 00666    0660  0762
WRITEPI   00006 0008F6 00761    0649
WRITEPS   00004 0008AC 00739    0731
```

3/22/72

SYMBOL	LEN	VALUE	DEFN	REFERENCES												
WRITE1	00006	000700	00602	0627												
WRITE2	00006	00057C	00488	0512												
WRITE3	00006	000434	00398	0423												
WRITE4	00006	00037A	00330	0355												
WRTFMT1	00002	0006FC	00600	0209	0595											
WRTFMT2	00002	000540	00474	0475												
WRTFMT3	00002	000426	00392	0393												
WRTFMT4	00002	000328	00301	0302												
WRTRSPNS	00002	000540	00475	0434	0441	0449	0457	0465	0469	0547	0587					
ZEROPARM	00002	000978	00804	0799												
ZEROREG	00001	000000	00028	0095	0096	0157	0158	0159	0370	0375	0524	0539	0540	0554	0555	0654
ZEROWRD	00004	000AC8	00845													

NO STATEMENTS FLAGGED IN THIS ASSEMBLY
STATISTICS SOURCE RECORDS (SYSIN) = 691 SOURCE RECORDS (SYSLIB) = 5465
OPTIONS IN EFFECT LIST, DECK, NOLOAD, NORENT, XREF, NOTEST, ALGN, OS, NOTERM, LINECNT = 55
1265 PRINTED LINES

The remote 3270 sample program, which exercises the remote 3270 display system, is distributed as a member of SYS1.SAMPLIB named SAMP327R. This PDS member contains (in the form of 80-character card images) all the source statements for the sample program and all the JCL needed to assemble, link-edit, and execute the program:

```
//SAMP327R JOB BTAM,MSGLEVEL=1
// EXEC ASMFCLG
//ASM.SYSIN DD *
         .
         .
         .
     Source Code
         .
         .
         .
/*
//GO.DD3270 DD UNIT=address of BSC line
//GO.SYSABEND DD SYSOUT=A
/*
```

Before assembling the program, supply the address of the BSC line in the 270X control unit as the UNIT parameter in the //GO.DD3270 DD card. Place the cards in the card reader, and perform the assemble, link-edit, and go procedure. The messages and instructions that appear on the screen of the display station are self-explanatory.

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                        F01OCT71    3/22/72
000000                                1 SAMP327R CSECT


                                      3 *        THIS IS A SAMPLE PROGRAM FOR A REMOTE 3270 INFORMATION DISPLAY
                                      4 *  SYSTEM.  PLEASE NOTE THAT THIS PROGRAM HAS BEEN WRITTEN FOR TWO
                                      5 *  3270 DEVICES, BOTH OF WHICH MUST BE 3277 DISPLAYS.
                                      6 *
                                      7 *        THIS PROGRAM IS SET UP FOR AN ASSEMBLE, LINKEDIT AND GO .
                                      8 *    THE DD3270 DD CARD WILL HAVE TO BE CHANGED TO REFLECT THE
                                      9 *    ADDRESS OF THE BSC LINE IN THE TRANSMISSION CONTROL UNIT .
                                     10 *
                                     11 *        IT MAY BE NECESSARY TO ALTER THIS SAMPLE PROBLEM SO THAT IT
                                     12 *  MAY FUNCTION WITH MORE THAN TWO DEVICES.  TO DO SO, THE FOLLOWING
                                     13 *  CARDS MUST BE CHANGED IN THE SOURCE DECK:
                                     14 *                          1. DFTRMLST MACROS  (SELDSPLY)
                                     15 *                                              (SPECPOL)
                                     16 *
                                     17 *        THIS SAMPLE PROGRAM IS RESTRICTED TO A MAXIMIM OF 32 DEVICES,
                                     18 *  ALL OF WHICH MUST BE ATTACHED TO ONE REMOTE 3271 CONTROL UNIT.
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71   3/22/72

                                     20 *        REGISTER EQUATES
000000                               21 REGZERO  EQU   0               REGISTER 0
000002                               22 REG2     EQU   2               WORK REGISTER
000003                               23 WORKREG  EQU   3               WORK REGISTER
000004                               24 MSGADDR  EQU   4               ADDR OF OUTPUT MESSAGE
000005                               25 MSGLEN   EQU   5               LENGTH OF OUTPUT MESSAGE
000006                               26 SELREG   EQU   6          SPEC POLL ADDR OF 3270 DISPLAY
000008                               27 FMTREG   EQU   8               FORMAT IDENTIFIER
000009                               28 DSPTABRG EQU   9               ADDRESS OF DISPLAY TABLE
00000A                               29 LNKREG   EQU   10              LINKAGE REGISTER
00000B                               30 BASEREG  EQU   11              FIRST BASE REG
00000C                               31 BASEREG2 EQU   12              SENOND BASE REG
00000D                               32 PTRTAB   EQU   13              PRINTER RLN TABLE ADDRESS
00000F                               33 RTNCDRG  EQU   15              RETURN CODE REGISTER


                                     35 *        EQUATES
000000                               36 ZERO     EQU   0               LENGTH OF 0
000001                               37 ONE      EQU   1               LENGTH OF 1
000002                               38 TWO      EQU   2               LENGTH OF 2
000003                               39 THREE    EQU   3               LENGTH OF 3
000004                               40 FOUR     EQU   4               FORMAT 1 IDENTIFIER
000005                               41 FIVE     EQU   5          LENGTH OF TERMINAL LIST ENTRIES
000006                               42 SIX      EQU   6               LENGTH OF 6
000006                               43 TP06     EQU   6               TP CODE OF 6
000007                               44 SEVEN    EQU   7               LENGTH OF 7
000008                               45 EIGHT    EQU   8               FORMAT 2 IDENTIFIER
000011                               46 TP11     EQU   X'11'           TP CODE OF 11
000010                               47 SIXTN    EQU   16              LENGTH IF 16
000020                               48 TP20     EQU   X'20'
000018                               49 TWENTY4  EQU   24         LENGTH OF 24
00001C                               50 TWENTY8  EQU   28              LENGTH OF 28
000001                               51 TIMEOUT  EQU   X'01'           DECB TIME OUT FLAG
000010                               52 SSMSG    EQU   X'10'           SENSE/STATUS RECEIVED FLAG
000040                               53 EOTRSPTX EQU   X'40'           EOT RESPONSE TO TEXT
000040                               54 EOTRCVD  EQU   X'40'           EOT RECEIVED FLAG
000041                               55 FOURTY1  EQU   X'41'           I/O ERROR COMP CODE
00006C                               56 PA1      EQU   X'6C'           ATTENTION ID FOR PA1 KEY
00006D                               57 CLEAR    EQU   X'6D'           ATTENTION ID FOR CLEAR KEY
00006E                               58 PA2      EQU   X'6E'           ATTENTION ID FOR PA2 (CNCL) KEY
00007D                               59 ENTER    EQU   X'7D'           ATTENTION ID FOR ENTER KEY
00007F                               60 SEVENF   EQU   X'7F'           NORMAL COMPLETION CODE
000080                               61 LAST     EQU   X'80'      SIGNIFIES END OF POLLING LIST
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    F01OCT71   3/22/72

                                    63           SAVE  (14,12)
000000                              64+          DS    0H
000000 90EC D00C            0000C   65+          STM   14,12,12(13) SAVE REGISTERS
000004 05B0                         66           BALR  BASEREG,0           ESTABLISH
000006                              67           USING *,BASEREG,BASEREG2            ADDRESSABILITY
000006 18CB                         68           LR    BASEREG2,BASEREG    INITIALIZE
000008 4AC0 B714            0071A   69           AH    BASEREG2,H4096                 SECOND BASE
00000C 50D0 BF1E            00F24   70           ST    13,SAVE+4
000010 41D0 BF1A            00F20   71           LA    13,SAVE
                                    72 *         OPEN THE LINE GROUP
                                    73           OPEN  (DCBR)    OPEN THE DCB
000014                              74+          CNOP  0,4
000014 4510 B016            0001C   75+          BAL   1,*+8 LOAD REG1 W/LIST ADDR.
000018 80                           76+          DC    AL1(128) OPTION BYTE
000019 000F58                       77+          DC    AL3(DCBR) DCB ADDRESS
00001C 0A13                         78+          SVC   19 ISSUE OPEN SVC
00001E 9110 BF82      00F88         79           TM    DCBR+48,X'10'      TEST FOR SUCCESSFUL OPEN
000022 4710 B03C            00042   80           BO    START          YES, GO TO START
                                    81           WTO   'OPEN FAILURE'
000026 0700                         82+          CNOP  0,4
000028 4510 B036            0003C   83+          BAL   1,IHB0003A BRANCH AROUND MESSAGE
00002C 0010                         84+          DC    AL2(16) TEXT LENGTH
00002E 0000                         85+          DC    B'0000000000000000' MCS FLAGS
000030 D6D7C5D540C6C1C9             86+          DC    C'OPEN FAILURE'
00003C                              87+IHB0003A DS     0H
00003C 0A23                         88+          SVC   35
00003E 47F0 B2C2            002C8   89           B     ABNORMAL
000042 4190 B71A            00720   90 START     LA    DSPTABRG,DSPTAB     ADDRESS THE DISPLAY TABLE
000046 4160 B6EA            006F0   91           LA    SELREG,SELDSPLY     GET SELECTION ADDRESS OF
                                    92 *                                           FIRST 3270 DISPLAY


00004A                              94 INITIAL    EQU  *
00004A 4140 B2DA            002E0   95           LA    MSGADDR,FORMAT0     ADDR OF FORMAT0 MESSAGE
00004E 4150 00A7            000A7   96           LA    MSGLEN,FMT0SZ       LENGTH OF MESSAGE
000052 45A0 B5C6            005CC   97           BAL   LNKREG,WRITETI      GO WRITE FORMAT 0
000056 45A0 B1D2            001D8   98           BAL   LNKREG,RETCODE      CHECK RETURN CODE
00005A 45A0 B222            00228   99           BAL   LNKREG,WAITD        WAIT FOR COMPLETION
00005E 4166 0005            00005   100          LA    SELREG,FIVE(SELREG) ADDR NEXT SPECIFIC POLL ENTRY
000062 9180 6000      00000         101          TM    ZERO(SELREG),LAST   END OF SELECTION LIST
000066 4710 B06C            00072   102          BO    READ                YES, GO ISSUE A READ
00006A 4166 0001            00001   103          LA    SELREG,ONE(SELREG)  ADDR OF NEXT ENTRY
00006E 47F0 B044            0004A   104          B     INITIAL             NO, WRITE TO REMAINING DISPLAYS


000072                              106 READ      EQU  *
000072 4140 B74A            00750   107          LA    MSGADDR,INAREA      ADDR OF INPUT AREA
000076 D7FE B74A B74A 00750 00750   108          XC    INAREA(255),INAREA  CLEAR INPUT
00007C D72A B74A B74A 00750 00750   109          XC    INAREA(43),INAREA              AREA
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                               F01OCT71   3/22/72

000082 45A0 B5FA            00600   110            BAL   LNKREG,READTI          GO READ A DISPLAY
000086 45A0 B1D2            001D8   111            BAL   LNKREG,RETCODE         CHECK RETURN CODE
00008A 45A0 B222            00228   112            BAL   LNKREG,WAITD           WAIT FOR COMPLETION
00008E D201 B73F B74C 00745 00752   113            MVC   CUDVSAVE(TWO),INAREA+TWO SAVE CU,DV
000094 9102 BFCE      00FD4         114            TM    DECBD+TWENTY4,X'02'              STATUS MSG RECIEVED
000098 4710 B2C2            002C8   115            BO    SSCHECK
00009C 4160 B6F6            006FC   116            LA    SELREG,SPECPOL         ADDR OD SPEC POLL TABLE
0000A0 1B22                         117            SR    REG2,REG2              CLEAR REGISTER 2
0000A2                              118 CHKIT      EQU   *
0000A2 D501 6001 B74C 00001 00752   119            CLC    ONE(TWO,SELREG),INAREA+TWO CHECK FOR CU,DV
0000A8 4780 B0BA            000C0   120            BE    FNDSEL                 YES, GET SELECTION ADDR
0000AC 4122 0001            00001   121            LA    REG2,ONE(REG2)         ADD ONE TO INDEX
0000B0 4166 0006            00006   122            LA    SELREG,SIX(SELREG)     POINT TO NEXT ENTRY
0000B4 9180 6000      00000         123            TM    ZERO(SELREG),LAST      END OF LIST
0000B8 4710 B2C2            002C8   124            BO    ABNORMAL
0000BC 47F0 B09C            000A2   125            B     CHKIT                  NO KEEP CHECKING
0000C0                              126 FNDSEL     EQU   *
0000C0 4220 B73E            00744   127            STC   REG2,INDEX             SAVE INDEX BYTE
0000C4 1832                         128            LR    WORKREG,REG2           GET INDEX INTO ODD REGISTER
0000C6 5C20 B746            0074C   129            M     REG2,SIXL              MULTIPLY INDEX BY 6
0000CA 1823                         130            LR    REG2,WORKREG           RE-ESTABLISH INDEX REG
0000CC 4160 B6EA            006F0   131            LA    SELREG,SELDSPLY        GET SELECTION ADDRES
0000D0 1A62                         132            AR    SELREG,REG2            ADDR OUTPUT ENTRY IN TABLE
0000D2 5060 B742            00748   133            ST    SELREG,SELSAVE         SAVE SELECTION ADDR
0000D6 1B88                         134            SR    FMTREG,FMTREG          CLEAR FORMAT REG
0000D8 4190 B71A            00720   135            LA    DSPTABRG,DSPTAB        ADDR OF DISPLAY TABLE
0000DC 4389 2000            00000   136            IC     FMTREG,ZERO(DSPTABRG,REG2) GET FORMAT ID
0000E0 47F8 B0DE            000E4   137            B     FORMATBR(FMTREG)


0000E4                              139 FORMATBR EQU     *
0000E4 47F0 B0EA            000F0   140            B     FMT0                   FORMAT 0 ON SCREEN
0000E8 47F0 B10A            00110   141            B     FMT1                   FORMAT 1 ON SCREEN
0000EC 47F0 B196            0019C   142            B     FMT2                   FORMAT 2 ON SCREEN


0000F0                              144 FMT0       EQU   *
                                    145 *   VERIFY THE NAME AND SOCIAL SECURITY NUMBER. ASSUMING THAT THEY
                                    146 * ARE VALID, WE SHALL CONTINUE PROCESSING.
0000F0                              147 FMT01      EQU   *
0000F0 4180 0004            00004   148            LA    FMTREG,FOUR            GET FORMAT 1 ID
0000F4 4289 2000            00000   149            STC    FMTREG,ZERO(DSPTABRG,REG2) STORE IN DISPLAY TABLE
0000F8 4140 B381            00387   150            LA    MSGADDR,FORMAT1        ADDR OF FORMAT1 MESSAGE
0000FC 4150 00DA            000DA   151            LA    MSGLEN,FMT1SZ          LENGTH OF MESSAGE
000100 45A0 B5C6            005CC   152            BAL    LNKREG,WRITETI        GO WRITE FORMAT 1
000104 45A0 B1D2            001D8   153            BAL   LNKREG,RETCODE         CHECK RETURN CODE
000108 45A0 B222            00228   154            BAL   LNKREG,WAITD           WAIT FOR COMPLETION
00010C 47F0 B06C            00072   155            B     READ                   GO READ ANOTHER DISPLAY
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                F01OCT71   3/22/72

000110                              157 FMT1     EQU   *
000110 957D B74E      00754         158          CLI   INAREA+FOUR,ENTER    ENTER KEY INTERRUPT
000114 4780 B12E            00134   159          BE    ENTERINT             YES, GO UPDATE RECORDS
000118 956C B74E      00754         160          CLI   INAREA+FOUR,PA1      PA1 KEY INTERRUPT
00011C 4780 B146            0014C   161          BE    PA1INT               YES, GO MAKE HARD COPY
000120 956E B74E      00754         162          CLI   INAREA+FOUR,PA2      PA2 OR CNCL KEY INTERRUPT
000124 4780 B14A            00150   163          BE    PA2INT               YES, GO DEACTIVATE TERMINAL
000128 956D B74E      00754         164          CLI   INAREA+FOUR,CLEAR    CLEAR KEY INTERRUPT
00012C 4780 B176            0017C   165          BE    CLEARINT             YES, GO WRITE FORMAT 2
000130 47F0 B06C            00072   166          B     READ                 IGNORE THE INTERRUPT AND GO READ


000134                              168 ENTERINT EQU   *
                                    169 *    CREATE A NEW OR UPDATE AN EXISTING ENTRY IN YOUR PERMANENT
                                    170 * DATA SET.
000134 4140 B5BF            005C5   171          LA    MSGADDR,ERALUNP      ADDR OF MESSAGE
000138 4150 0004            00004   172          LA    MSGLEN,ERALUNPL      LENGTH OF MESSAGE
00013C 45A0 B5C6            005CC   173          BAL   LNKREG,WRITETI       GO ERASE ALL UNPROTECTED DATA
000140 45A0 B1D2            001D8   174          BAL   LNKREG,RETCODE       CHECK RETURN CODE
000144 45A0 B222            00228   175          BAL   LNKREG,WAITD         WAIT FOR COMPLETION
000148 47F0 B06C            00072   176          B     READ                 GO READ ANOTHER DISPLAY
00014C                              177 PA1INT   EQU   *
00014C 47F0 B1BA            001C0   178          B     NOPRINT              NO PRINTER DEFINED
000150                              179 PA2INT   EQU   *
                                    180 *    DETERMINE IF ANY DATA WAS ENTERED. IF SO, CREATE A NEW OR UPDATE
                                    181 * AN EXISTING ENTRY IN YOUR PERMANENT DATA SET. NOW DEACTIVATE THE
                                    182 * TERMINAL.
000150                              183 PA2INT1  EQU   *
000150 4140 B57F            00585   184          LA    MSGADDR,CLOSEMG      ADDR OF CLOSE MSG
000154 4150 003C            0003C   185          LA    MSGLEN,CLOSEMGL      LENGTH OF MSG
000158 4160 B6EA            006F0   186          LA    SELREG,SELDSPLY      ADDR OF SELECTION TABLE
00015C                              187 ONCEMORE EQU   *
00015C 45A0 B5C6            005CC   188          BAL   LNKREG,WRITETI       GO WRITE ENDING MSG
000160 45A0 B1D2            001D8   189          BAL   LNKREG,RETCODE       CHECK RETURN CODE
000164 45A0 B222            00228   190          BAL   LNKREG,WAITD         WAIT FOR COMPLETION
000168 4166 0005            00005   191          LA    SELREG,FIVE(SELREG)  POINT TO INDICATOR BYTE
00016C 9180 6000      00000         192          TM    ZERO(SELREG),LAST    END OF SELECTION LIST
000170 4710 B2AE            002B4   193          BO    CLOSE                YES, TERMINATE PROGRAM
000174 4166 0001            00001   194          LA    SELREG,ONE(SELREG)   POINT TO NEXT ADDRESS
000178 47F0 B156            0015C   195          B     ONCEMORE             NO, WRITE ANOTHER MESSAGE
00017C                              196 CLEARINT EQU   *
00017C 4140 B45B            00461   197          LA    MSGADDR,FORMAT2      ADDR OF FORMAT 2 MSG
000180 4150 00F7            000F7   198          LA    MSGLEN,FMT2SZ        LENGTH OF MSG
000184 45A0 B5C6            005CC   199          BAL   LNKREG,WRITETI       GO WRITE FORMAT 2
000188 45A0 B1D2            001D8   200          BAL   LNKREG,RETCODE       CHECK RETURN CODE
00018C 45A0 B222            00228   201          BAL   LNKREG,WAITD         WAIT FOR COMPLETION
000190 4180 0008            00008   202          LA    FMTREG,EIGHT         GET FORMAT 2 ID
000194 4289 2000            00000   203          STC   FMTREG,ZERO(DSPTABRG,REG2) STORE IN DISPLAY TABLE
000198 47F0 B06C            00072   204          B     READ                 GO READ ANOTHER DISPLAY
00019C                              205 FMT2     EQU   *
00019C 957D B74E      00754         206          CLI   INAREA+FOUR,ENTER    ENTER KEY INTERRUPT
0001A0 4780 B0EA            000F0   207          BE    FMT01                YES, GO WRITE FORMAT 1
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71   3/22/72

0001A4 956C B74E      00754         208          CLI    INAREA+FOUR,PA1     PA1 KEY INTERRUPT
0001A8 4780 B146            0014C   209          BE     PA1INT              YES, GO MAKE HARD COPY
0001AC 956E B74E      00754         210          CLI    INAREA+FOUR,PA2     PA2 OR CNCL KEY INTERRUPT
0001B0 4780 B14A            00150   211          BE     PA2INT1             YES, GO DEACTIVATE TERMINAL
0001B4 956D B74E      00754         212          CLI    INAREA+FOUR,CLEAR   CLEAR KEY INTERRUPT
0001B8 4780 B176            0017C   213          BE     CLEARINT            GO WRITE FORMAT 2
0001BC 47F0 B06C            00072   214          B      READ                GO READ ANOTHER DISPLAY
0001C0                              215 NOPRINT  EQU    *
0001C0 4140 B552            00558   216          LA     MSGADDR,NOPTR       ADDR OF NO PRINTER MSG
0001C4 4150 002D            0002D   217          LA     MSGLEN,NOPTRL       LENGTH OF MSG
0001C8 45A0 B5C6            005CC   218          BAL    LNKREG,WRITETI      GO WRITE MSG
0001CC 45A0 B1D2            001D8   219          BAL    LNKREG,RETCODE      CHECK RETURN CODE
0001D0 45A0 B222            00228   220          BAL    LNKREG,WAITD        WAIT FOR COMPLETION
0001D4 47F0 B06C            00072   221          B      READ                GO READ ANOTHER DISPLAY
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F01OCT71   3/22/72

0001D8                              223 RETCODE  EQU   *
0001D8 47FF B1D6             001DC  224          B     RTNCDTAB(RTNCDRG)   BRANCH TO CORRESPONDING ENYRY
0001DC                              225 RTNCDTAB EQU   *
0001DC 47F0 B1FE             00204  226          B     RTNCD0              I/O SUCCESSFULLY INITIATED
0001E0 47F0 B200             00206  227          B     RTNCD4              DTFBT BUSY
0001E4 47F0 B206             0020C  228          B     RTNCD8              INVALID RLN
0001E8 47F0 B206             0020C  229          B     RTNCDC              INVALID TYPE CODE
0001EC 47F0 B20A             00210  230          B     RTNCD10             ALL SKIP BITS ON
0001F0 47F0 B20E             00214  231          B     RTNCD14             LINE ERROR AT OPEN
0001F4 47F0 B212             00218  232          B     RTNCD18             NO BUFFERS
0001F8 47F0 B216             0021C  233          B     RTNCD1C             NO BUFFER POOL
0001FC 47F0 B21A             00220  234          B     RTNCD20             NO BUFFER MANAGEMENT
000200 47F0 B21E             00224  235          B     RTNCD24             BSC USAGE COUNT EXCEEDED
000204                              236 RTNCD0   EQU   *
000204 07FA                         237          BR    LNKREG              RETURN
000206                              238 RTNCD4   EQU   *
000206 5BA0 B716             0071C  239          S     LNKREG,EIGHT8       SUBTRACT 8 FROM RETURN ADDR
00020A 07FA                         240          BR    LNKREG                 TO RETRY THE OPERATION
00020C                              241 RTNCD8   EQU   *
00020C                              242 RTNCDC   EQU   *
00020C 47F0 B2C2             002C8  243          B     ABNORMAL            THIS CONDITION SHOULD NOT OCCUR
000210                              244 RTNCD10  EQU   *
000210 47F0 B2AE             002B4  245          B     CLOSE               ALL TERMINALS, TERMINATE
000214                              246 RTNCD14  EQU   *
000214 47F0 B2C2             002C8  247          B     ABNORMAL            THIS CONDITION SHOULD NOT OCCUR
000218                              248 RTNCD18  EQU   *
000218 47F0 B2C2             002C8  249          B     ABNORMAL            THIS CONDITION SHOULD NOT OCCUR
00021C                              250 RTNCD1C  EQU   *
00021C 47F0 B2C2             002C8  251          B     ABNORMAL            THIS CONDITION SHOULD NOT OCCUR
000220                              252 RTNCD20  EQU   *
000220 47F0 B2C2             002C8  253          B     ABNORMAL            THIS CONDITION SHOULD NOT OCCUR
000224                              254 RTNCD24  EQU   *
000224 47F0 B2C2             002C8  255          B     ABNORMAL            THIS CONDITION SHOULD NOT OCCUR
```

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F01OCT71   3/22/72

000228                              257 WAITD    EQU   *
                                    258          WAIT  ECB=DECBD
000228 4110 BFB6            00FBC   259+         LA    1,DECBD LOAD PARAMETER REG 1
00022C 4100 0001            00001   260+         LA    0,1(0,0) COUNT OMITTED,1 USED
000230 0A01                         261+         SVC   1 LINK TO WAIT ROUTINE
000232                              262 CHK7FCC  EQU   *
000232 957F BFB6      00FBC         263          CLI   DECBD,SEVENF        NORMAL COMPLETION CODE
000236 4770 B236            0023C   264          BNE   CHK41CC             NO, KEEP CHECKING
00023A 07FA                         265          BR    LNKREG              RETURN
00023C                              266 CHK41CC  EQU   *
00023C 9541 BFB6      00FBC         267          CLI   DECBD,FOURTY1       I/O ERROR COMP CODE
000240 4770 B2C2            002C8   268          BNE   ABNORMAL          INVALID COMP CODE
000244 9506 BFD2      00FD8         269          CLI   DECBD+TWENTY8,TP06  TP CODE OF 6
000248 4770 B254            0025A   270          BNE   CKTP20              NO, CHECK OTHER TP CODES
00024C 9101 BFC6      00FCC         271          TM    DECBD+SIXTN,TIMEOUT DID DEVICE TIME OUT
000250 47E0 B2C2            002C8   272          BNO   ABNORMAL            NO, TERMINATE
000254 5BA0 B73A            00740   273          S     LNKREG,TWELVE       SUBTRACR 12 FROM RETURN ADDR
000258 07FA                         274          BR    LNKREG              TO RETRY THE OPERATION
00025A                              275 CKTP20   EQU   *
00025A 9520 BFD2      00FD8         276          CLI   DECBD+TWENTY8,TP20  TP CODE OF 20
00025E 4770 B268            0026E   277          BNE   CKTP11              NO, CHECK OTHERS
000262                              278 CHKEOT   EQU   *
000262 9140 BFCE      00FD4         279          TM    DECBD+TWENTY4,EOTRSPTX EOT RESPONSE TO TEXT
000266 47E0 B2C2            002C8   280          BNO   ABNORMAL            NO, TERMINATE
00026A 47F0 B06C            00072   281          B     READ                YES, GO GET SENSE STATUS MESSAGE
00026E                              282 CKTP11   EQU   *
00026E 9511 BFD2      00FD8         283          CLI   DECBD+TWENTY8,TP11  TP CODE OF 11
000272 4770 B2C2            002C8   284          BNE   ABNORMAL            NO, TERMINATE
000276 47F0 B25C            00262   285          B     CHKEOT              YES, CHECK EOT RESPONSE TO TEXT
00027A                              286 CHKTP20  EQU   *
00027A 9520 BFD2      00FD8         287          CLI   DECBD+TWENTY8,TP20  TP CODE OF 20
00027E 4770 B2C2            002C8   288          BNE   ABNORMAL            NO, TERMINATE
                                    289          WRITE DECBD,TR,DCBR,INAREA,,,1,MF=E
000282 4110 BFB6            00FBC   290+         LA    1,DECBD LOAD DECB ADDRESS
000286 9405 1004      00004         291+         NI    4(1),5
00028A 920A 1005      00005         292+         MVI   5(1),10
00028E 41E0 BF52            00F58   293+         LA    14,DCBR
000292 50E0 1008            00008   294+         ST    14,8(0,1) STORE DCB ADDRESS
000296 41E0 B74A            00750   295+         LA    14,INAREA 000B
00029A 50E0 100C            0000C   296+         ST    14,12(0,1) STORE AREA ADDR    000B
00029E 41E0 0001            00001   297+         LA    14,1(0,0)
0002A2 40E0 1018            00018   298+         STH   14,24(0,1) STORE LINE NUMBER
0002A6 58F0 BF82            00F88   299+         L     15,DCBR+48 LOAD RDWRT ROUT ADDR
0002AA 94F7 1004      00004         300+         NI    4(1),X'F7' A38557
0002AE 05EF                         301+         BALR  14,15 A38557
0002B0 47F0 B22C            00232   302          B     CHK7FCC             CHECK COMPLETION CODE
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                             F01OCT71   3/22/72

0002B4                             304 CLOSE    EQU   *
                                   305          CLOSE (DCBR)    CLOSE THE DCB
0002B4                             306+         CNOP  0,4
0002B4 4510 B2B6            002BC  307+         BAL   1,*+8 BRANCH AROUND LIST
0002B8 80                          308+         DC    AL1(128) OPTION BYTE
0002B9 000F58                      309+         DC    AL3(DCBR) DCB ADDRESS
0002BC 0A14                        310+         SVC   20 ISSUE CLOSE SVC
0002BE 58D0 BF1E            00F24  311          L    13,SAVE+4
                                   312          RETURN (14,12)
0002C2 98EC D00C            0000C  313+         LM    14,12,12(13) RESTORE THE REGISTERS
0002C6 07FE                        314+         BR    14 RETURN


0002C8                             316 SSCHECK  EQU   *
                                   317 *     INVESTIGATE THE SENSE/STATUS BYTES SENT BY THE REMOTE DEVICE.
                                   318 *  IF RECOVERY IS POSSIBLE, ATTEMPT TO DO SO.  WE SHALL ASSUME THAT
                                   319 *  THE ERROR IS UNRECOVERABLE AND TERMINATE.


0002C8                             321 ABNORMAL EQU   *
                                   322          ABEND 1,DUMP      TAKE A DUMP
0002C8                             323+         DS    0H
0002C8                             324+         CNOP  0,4
0002C8 47F0 B2CA            002D0  325+         B     *+8 BRANCH AROUND CONSTANT
0002CC 80                          326+         DC    AL1(128) DUMP/STEP CODE
0002CD 000001                      327+         DC    AL3(1) COMPLETION CODE
0002D0 5810 B2C6            002CC  328+         L     1,*-4 LOAD CODES INTO REG 1
0002D4 0A0D                        329+         SVC   13 LINK TO ABEND ROUTINE
0002D6 58D0 BF1E            00F24  330          L    13,SAVE+4
                                   331          RETURN (14,12)
0002DA 98EC D00C            0000C  332+         LM    14,12,12(13) RESTORE THE REGISTERS
0002DE 07FE                        333+         BR    14 RETURN
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                               F01OCT71    3/22/72

0002E0                               335 FORMAT0   EQU    *
0002E0 0227F5                        336           DC     X'0227F5'              STX,ESC,E/W
0002E3 C71DC811C150                  337           DC     X'C71DC811C150'        WCC, SF = PROT, SBA = 80
0002E9 C7D6D6C440D4D6D9              338           DC     C'GOOD MORNING.'
0002F6 1D6011C15F                    339           DC     X'1D6011C15F'          SF = PROT, SBA = 94
0002FB E3C8C9E240C2C5C7              340           DC     C'THIS BEGINS THE DEMONSTRATION '
000319 D6C640E3C8C54040              341           DC     C'OF THE  OS/BTAM '
000329 1DC8                          342           DC     X'1DC8'                SF = PROT
00032B F3F2F7F040                    343           DC     C'3270 '
000330 1D60                          344           DC     X'1D60'                SF = PROT
000332 D9C5D4D6E3C540E2              345           DC     C'REMOTE SAMPLE PROGRAM.'
000348 1D6011C3F0                    346           DC     X'1D6011C3F0'          SF = PROT, SBA = 240
00034D C5D5E3C5D940E3C8              347           DC     C'ENTER THE FOLLOWING:'
000361 1D6011C4D8                    348           DC     X'1D6011C4D8'          SF = PROT, SBA = 280
000366 D5C1D4C57A                    349           DC     C'NAME:'
00036B 1D4013                        350           DC     X'1D4013'              SF = UNPROT, IC
00036E 11C5401D60                    351           DC     X'11C5401D60'          SBA = 320,SF = PROT
000373 E2D6C340E2C5C340              352           DC     C'SOC SEC NUM:'
00037F 1D40                          353           DC     X'1D40'                SF = UNPROT
000381 11C5E81D60                    354           DC     X'11C5E81D60'          SBA = 360,SF = PROT
000386 03                            355           DC     X'03'                  ETX
0000A7                               356 FMT0SZ    EQU    *-FORMAT0


000387                               358 FORMAT1   EQU    *
000387 0227F5                        359           DC     X'0227F5'              STX,ESC,E/W
00038A C71D60114040                  360           DC     X'C71D60114040'        WCC, SF = PROT, SBA = 0
000390 C5D5E3C5D940C4C1              361           DC     C'ENTER DATA REQUESTED BELOW:'
0003AB 11C150                        362           DC     X'11C150'              SBA = 80
0003AE D5C1D4C57A                    363           DC     C'NAME:'
0003B3 1D401311C1F81D60              364           DC     X'1D401311C1F81D60'    SF = UNPROT, IC, SBA = 120,
                                     365 *                                        SF = PROT
0003BB C1C4C4D97A                    366           DC     C'ADDR:'
0003C0 1D4011C2601D60                367           DC     X'1D4011C2601D60'      SF = UNPROT, SBA = 160, SF = PROT
0003C7 C3C9E3E87A                    368           DC     C'CITY:'
0003CC 1D4011C3C81D60                369           DC     X'1D4011C3C81D60'      SF = UNPROT, SBA = 200, SF = PROT
0003D3 E2E3C1E3C57A                  370           DC     C'STATE:'
0003D9 1D4011C3E41D60                371           DC     X'1D4011C3E41D60'      SF = UNPROT, SBA = 228, SF = PROT
0003E0 E9C9D77A                      372           DC     C'ZIP:'
0003E4 1D5011C3F01D6011              373           DC     X'1D5011C3F01D6011C4D8'  SF = UNPROT, SBA = 240,
                                     374 *                                        SF = PROT, SBA = 280
0003EE C5D5E3C5D940D2C5              375           DC     C'ENTER KEY: ENTER DATA;'
000404 11C540                        376           DC     X'11C540'              SBA = 320
000407 D7C1F140D2C5E87A              377           DC     C'PA1 KEY: PRINT DATA;'
00041B 11C5E8                        378           DC     X'11C5E8'              SBA = 360
00041E D7C1F2404DC3D5C3              379           DC     C'PA2 (CNCL) KEY: DEACTIVATE TERMINAL;'
000442 11C650                        380           DC     X'11C650'              SBA = 400
000445 C3D3C5C1D940D2C5              381           DC     C'CLEAR KEY: CONTROL OPTIONS;'
000460 03                            382           DC     X'03'                  ETX
0000DA                               383 FMT1SZ    EQU    *-FORMAT1
```

```
  LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F01OCT71   3/22/72

000461                               385 FORMAT2  EQU   *
000461 0227F5                        386          DC    X'0227F5'           STX,ESC,E/W
000464 C711404013                    387          DC    X'C711404013'       WCC, SBA = 0, IC
000469 E7E7E8E8F4C3C3C4              388          DC    C'XXYY4CCDD'
000472 1140E8                        389          DC    X'1140E8'           SBA = 40
000475 E3D640D9C5D8E4C5              390          DC    C'TO REQUEST BTAM OLT -- ENTER REQUEST FOR TEST MESS'
0004A7 C1C7C540D6E5C5D9              391          DC    C'AGE OVER SAMPLE FORMAT ABOVE:  XX=TEST NO. (23-28)'
0004D9 4040E8E87ED9C5D7              392          DC    C'  YY=REPEATS (01-99) CCDD=ADDRESS OF TARGET DEVICE'
00050B E3C8C5D540C8C9E3              393          DC    C'THEN HIT ERASE EOF AND THEN TEST REQ. USE CLEAR KE'
00053D E840E3D640D9C5E2              394          DC    C'Y TO RESUME AFTER TEST.'
000554 11C3C8                        395          DC    X'11C3C8'           SBA = 200
000557 03                            396          DC    X'03'               ETX
0000F7                               397 FMT2SZ   EQU   *-FORMAT2


000558                               399 NOPTR    EQU   *
000558 0227F1                        400          DC    X'0227F1'           STX,ESC,WRITE
00055B C61DC811C6F8                  401          DC    X'C61DC811C6F8'     WCC, SF = PROT, SBA = 440
000561 D5D640D7D9C9D5E3              402          DC    C'NO PRINTER DEFINED FOR THIS PROGRAM'
000584 03                            403          DC    X'03'               ETX
00002D                               404 NOPTRL   EQU   *-NOPTR


000585                               406 CLOSEMG  EQU   *
000585 0227F5                        407          DC    X'0227F5'           STX,ESC,E/W
000588 C71140401DC8                  408          DC    X'C71140401DC8'     WCC, SBA = 0, SF = PROT
00058E 4040404040E3C8C5              409          DC    C'     THE REMOTE 3270 SAMPLE PROGRAM HAS CONCLUDED.'
0005C0 03                            410          DC    X'03'               ETX
00003C                               411 CLOSEMGL EQU   *-CLOSEMG


0005C1                               414 READBUF  EQU   *
0005C1 0227F203                      415          DC    X'0227F203'         STX,ESC,RD BUF,ETX
000004                               416 READBUFL EQU   *-READBUF


0005C5                               418 ERALUNP  EQU   *
0005C5 02276F03                      419          DC    X'02276F03'         STX,ESC,EAU,ETX
```



```
  LOC  OBJECT CODE      ADDR1 ADDR2  STMT   SOURCE STATEMENT                                          F01OCT71   3/22/72

000004                               420 ERALUNPL EQU   *-ERALUNP
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                             F01OCT71    3/22/72

                                   422 *        READ AND WRITE MACROS
0005CC                             423          DS    0F
0005CC                             424 WRITETI  EQU   *
                                   425          WRITE DECBD,TI,DCBR,(MSGADDR),(MSGLEN),(SELREG),1,MF=E
0005CC 4110 BFB6             00FBC 426+         LA    1,DECBD LOAD DECB ADDRESS
0005D0 9200 1004       00004       427+         MVI   4(1),0
0005D4 9202 1005       00005       428+         MVI   5(1),2
0005D8 4050 1006             00006 429+         STH   MSGLEN,6(0,1) STORE LENGTH        000B
0005DC 41E0 BF52             00F58 430+         LA    14,DCBR
0005E0 50E0 1008             00008 431+         ST    14,8(0,1) STORE DCB ADDRESS
0005E4 5040 100C             0000C 432+         ST    MSGADDR,12(0,1) STORE AREA ADDR   000B
0005E8 5060 1014             00014 433+         ST    SELREG,20(0,1) STORE TERM. LIST ADDR.
0005EC 41E0 0001             00001 434+         LA    14,1(0,0)
0005F0 40E0 1018             00018 435+         STH   14,24(0,1) STORE LINE NUMBER
0005F4 58F0 BF82             00F88 436+         L     15,DCBR+48 LOAD RDWRT ROUT ADDR
0005F8 94F7 1004       00004       437+         NI    4(1),X'F7' A38557
0005FC 05EF                        438+         BALR  14,15 A38557
0005FE 07FA                        439          BR    LNKREG


000600                             441 READTI   EQU   *
                                   442          READ  DECBD,TI,DCBR,INAREA,300,POLDSPLY,1,MF=E
000600 4110 BFB6             00FBC 443+         LA    1,DECBD LOAD DECB ADDRESS
000604 9200 1004       00004       444+         MVI   4(1),0
000608 9201 1005       00005       445+         MVI   5(1),1
00060C 41E0 012C             0012C 446+         LA    14,300(0,0) 000B
000610 40E0 1006             00006 447+         STH   14,6(0,1) STORE LENGTH       000B
000614 41E0 BF52             00F58 448+         LA    14,DCBR
000618 50E0 1008             00008 449+         ST    14,8(0,1) STORE DCB ADDRESS
00061C 41E0 B74A             00750 450+         LA    14,INAREA 000B
000620 50E0 100C             0000C 451+         ST    14,12(0,1) STORE AREA ADDR   000B
000624 41E0 B705             0070B 452+         LA    14,POLDSPLY
000628 50E0 1014             00014 453+         ST    14,20(0,1) STORE TERM. LIST ADDR.
00062C 41E0 0001             00001 454+         LA    14,1(0,0)
000630 40E0 1018             00018 455+         STH   14,24(0,1) STORE LINE NUMBER
000634 58F0 BF82             00F88 456+         L     15,DCBR+48 LOAD RDWRT ROUT ADDR
000638 94F7 1004       00004       457+         NI    4(1),X'F7' A38557
00063C 05EF                        458+         BALR  14,15 A38557
00063E 07FA                        459          BR    LNKREG
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71  3/22/72

000640                             461 READTRV  EQU   *
                                   462          READ  DECBD,TRV,DCBR,INAREA,256,,1,MF=E
000640 4110 BFB6             00FBC 463+         LA    1,DECBD LOAD DECB ADDRESS
000644 9404 1004       00004       464+         NI    4(1),4
000648 921B 1005       00005       465+         MVI   5(1),27
00064C 41E0 0100             00100 466+         LA    14,256(0,0) 000B
000650 40E0 1006             00006 467+         STH   14,6(0,1) STORE LENGTH        000B
000654 41E0 BF52             00F58 468+         LA    14,DCBR
000658 50E0 1008             00008 469+         ST    14,8(0,1) STORE DCB ADDRESS
00065C 41E0 B74A             00750 470+         LA    14,INAREA 000B
000660 50E0 100C             0000C 471+         ST    14,12(0,1) STORE AREA ADDR    000B
000664 41E0 0001             00001 472+         LA    14,1(0,0)
000668 40E0 1018             00018 473+         STH   14,24(0,1) STORE LINE NUMBER
00066C 58F0 BF82             00F88 474+         L     15,DCBR+48 LOAD RDWRT ROUT ADDR
000670 94F7 1004       00004       475+         NI    4(1),X'F7' A38557
000674 05EF                        476+         BALR  14,15 A38557
000676 07FA                        477          BR    LNKREG                 RETURN


000678                             479 WRITETIV EQU   *
                                   480          WRITE DECBD,TIV,DCBR,(INAREA,(MSGADDR)),(256,(MSGLEN)),          X
                                                      (SELREG),1,MF=E
000678 4110 BFB6             00FBC 481+         LA    1,DECBD LOAD DECB ADDRESS
00067C 9200 1004       00004       482+         MVI   4(1),0
000680 920D 1005       00005       483+         MVI   5(1),13
000684 41E0 0100             00100 484+         LA    14,256(0,0) 000B
000688 40E0 1006             00006 485+         STH   14,6(0,1) STORE LENGTH        000B
00068C 4050 102A             0002A 486+         STH   MSGLEN,42(0,1) STORE WLENGTH       000B
000690 41E0 BF52             00F58 487+         LA    14,DCBR
000694 50E0 1008             00008 488+         ST    14,8(0,1) STORE DCB ADDRESS
000698 41E0 B74A             00750 489+         LA    14,INAREA 000B
00069C 50E0 100C             0000C 490+         ST    14,12(0,1) STORE AREA ADDR    000B
0006A0 5040 102C             0002C 491+         ST    MSGADDR,44(0,1) STORE WAREA ADDR  000B
0006A4 5060 1014             00014 492+         ST    SELREG,20(0,1) STORE TERM. LIST ADDR.
0006A8 41E0 0001             00001 493+         LA    14,1(0,0)
0006AC 40E0 1018             00018 494+         STH   14,24(0,1) STORE LINE NUMBER
0006B0 58F0 BF82             00F88 495+         L     15,DCBR+48 LOAD RDWRT ROUT ADDR
0006B4 94F7 1004       00004       496+         NI    4(1),X'F7' A38557
0006B8 05EF                        497+         BALR  14,15 A38557
0006BA 07FA                        498          BR    LNKREG                 RETURN
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                            F01OCT71    3/22/72

0006BC                              500 READTT    EQU   *
                                    501           READ  DECBD,TT,DCBR,(MSGADDR),256,,1,MF=E
0006BC 4110 BFB6             00FBC  502+          LA    1,DECBD LOAD DECB ADDRESS
0006C0 9404 1004       00004        503+          NI    4(1),4
0006C4 9203 1005       00005        504+          MVI   5(1),3
0006C8 41E0 0100             00100  505+          LA    14,256(0,0) 000B
0006CC 40E0 1006             00006  506+          STH   14,6(0,1) STORE LENGTH       000B
0006D0 41E0 BF52             00F58  507+          LA    14,DCBR
0006D4 50E0 1008             00008  508+          ST    14,8(0,1) STORE DCB ADDRESS
0006D8 5040 100C             0000C  509+          ST    MSGADDR,12(0,1) STORE AREA ADDR   000B
0006DC 41E0 0001             00001  510+          LA    14,1(0,0)
0006E0 40E0 1018             00018  511+          STH   14,24(0,1) STORE LINE NUMBER
0006E4 58F0 BF82             00F88  512+          L     15,DCBR+48 LOAD RDWRT ROUT ADDR
0006E8 94F7 1004       00004        513+          NI    4(1),X'F7' A38557
0006EC 05EF                         514+          BALR  14,15 A38557
0006EE 07FA                         515           BR    LNKREG              RETURN
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                 F01OCT71   3/22/72

                                   517 *         DISPLAY SELECTION ADDRESSES
                                   518 *    THE CURRENT MACRO OPERANDS ARE FOR TWO REMOTE 3270 DISPLAYS;
                                   519 *                           1. 0TH CU, 0TH DEV (60604040)
                                   520 *                           2. 0TH CU, 1ST DEV (6060C1C1)
                                   521 SELDSPLY DFTRMLST OPENLST,(606040402D,6060C1C12D)
0006F0                             522+SELDSPLY DS     0X A28630
0006F0 606040402D                  523+         DC     X'606040402D' TERMINAL LIST ENTRY
0006F5 01                          524+         DC     AL1(1) PROCEDURE FLAGS
0006F6 6060C1C12D                  525+         DC     X'6060C1C12D' TERMINAL LIST ENTRY
0006FB 82                          526+         DC     AL1(130) PROCEDURE FLAGS


                                   528 *     DISPLAY SPECIFIC POLLING ADDRESSES
                                   529 *    THE CURRENT MACRO OPERANDS ARE FOR TWO REMOTE 3270 DISPLAYS:
                                   530 *                           1. 0TH CU, 0TH DEV (40404040)
                                   531 *                           2. 0TH CU, 1ST DEV (4040C1C1)
                                   532 SPECPOL  DFTRMLST OPENLST,(404040402D,4040C1C12D)
0006FC                             533+SPECPOL  DS     0X A28630
0006FC 404040402D                  534+         DC     X'404040402D' TERMINAL LIST ENTRY
000701 01                          535+         DC     AL1(1) PROCEDURE FLAGS
000702 4040C1C12D                  536+         DC     X'4040C1C12D' TERMINAL LIST ENTRY
000707 82                          537+         DC     AL1(130) PROCEDURE FLAGS
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71    3/22/72

                                   539 *         DISPLAY GENERAL POLLING ADDRESS
                                   540 POLDSPLY DFTRMLST AUTOWLST,(40407F7F2D,3737373737)
000708 0202                        541+          DC    2YL1(2) TOTAL AND ACTIVE ENTRY COUN000A
00070A D0                          542+          DC    YL1(X'20'*6+X'10'*1) PACKED WIDTH AND WRAP BIT  000A
00070B                             543+POLDSPLY DS     0X 000A
00070B 40407F7F2D                  544+          DC    XL5'40407F7F2D'
000710 01                          545+          DC    YL1(1) 000A
000711 3737373737                  546+          DC    XL5'3737373737'
000716 02                          547+          DC    YL1(2) 000A
000717 FE                          548+          DC    X'FE' END OF LIST CHARACTER       000A
000718 0010                        549+          DC    YL2(2*6+4) OFFSET                         000A

                                   552 *         CONSTANTS
00071A 1000                        553 H4096     DC    H'4096'
00071C 00000008                    554 EIGHT8    DC    F'8'                 CONSTANT OF 8
000720 0000000000000000            555 DSPTAB    DC    XL32'0'              DISPLAY TABLE
000740 0000000C                    556 TWELVE    DC    F'12'                LENGTH OF 12
000744 00                          557 INDEX     DC    X'00'                INDEX BYTE SAVE AREA
000745 0000                        558 CUDVSAVE DC     X'0000'              CU,DV SAVEAREA
000747 00
000748 00000000                    559 SELSAVE   DC    F'0'
00074C 00000006                    560 SIXL      DC    F'6'
000750 0000000000000000            561 INAREA    DC    500F'0'              INPUT AREA
0007D0                             562 INAREAL   EQU   *-INAREA
000F20 0000000000000000            563 SAVE      DC    18F'0'
000F68                             564           DS    0F
000F68 80                          565 DECBADDR DC     X'80'
000F69 000FBC                      566           DC    AL3(DECBD)
                                   567 DCBR      DCB   DSORG=CX,MACRF=(R,W),DDNAME=DD3270,EROPT=T,DEVD=BS,       X
                                                       MODE=(,,A,A),CODE=EBCDIC

                                   569+*                      DATA CONTROL BLOCK
                                   570+*
000F58                             571+          ORG   *-20 TO ELIMINATE UNUSED SPACE
000F58                             572+DCBR      DS    0F'0' ORIGIN ON WORD BOUNDARY
000F6C                             573+          ORG   *+20 TO ORIGIN GENERATION

                                   575+*                      COMMON ACCESS METHOD INTERFACE

000F6C 00                          577+          DC    AL1(0) BUFNO
000F6D 000001                      578+          DC    AL3(1) BUFCB
000F70 0000                        579+          DC    AL2(0) BUFL
000F72 1000                        580+          DC    BL2'0001000000000000' DSORG
000F74 00000001                    581+          DC    A(1) IOBAD
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                              F01OCT71   3/22/72

                                    583+*                     FOUNDATION EXTENSION

000F78 00                           585+          DC    BL1'00000000' BFTEK,BFLN,HIARCHY
000F79 10                           586+          DC    BL1'00010000' BTAM EROPT CODE
000F7A FF                           587+          DC    AL1(255) BTAM BUFFER COUNT
000F7B 00                           588+          DC    AL1(0)
000F7C 00                           589+          DC    BL1'00000000' RECFM
000F7D 000000                       590+          DC    AL3(0) EXLST

                                    592+*                     FOUNDATION BLOCK

000F80 C4C4F3F2F7F04040             594+          DC    CL8'DD3270' DDNAME
000F88 02                           595+          DC    BL1'00000010' OFLGS
000F89 00                           596+          DC    BL1'00000000' IFLG
000F8A 2020                         597+          DC    BL2'0010000000100000' MACR

                                    599+*                     BTAM INTERFACE

000F8C 00000000                     601+          DC    F'0'
000F90 00                           602+          DC    BL1'00000000' MODE
000F91 80                           603+          DC    BL1'10000000' MAS,CODE
000F92 0000000000000000             604+          DC    XL26'0' CONTROL CHARS
000FAC 0000000000000000             605+          DC    4F'0' RESERVED

                                    607           READ  DECBD,TI,DCBR,MF=L
000FBC                              608+          DS    0F
000FBC 00000000                     609+DECBD     DC    A(0) EVENT CONTROL BLOCK
000FC0 00                           610+          DC    BL1'000'
000FC1 01                           611+          DC    AL1(1) TYPE FIELD
000FC2 0000                         612+          DC    AL2(0) LENGTH
000FC4 00000F58                     613+          DC    A(DCBR) DCB ADDRESS
000FC8 00000000                     614+          DC    A(0) AREA ADDRESS
000FCC 00000000                     615+          DC    A(0) ERROR INFO. FIELD ADDR
000FD0 00000000                     616+          DC    A(0) TERMINAL LIST ADDRESS
000FD4 0000                         617+          DC    AL2(0) LINE NUMBER
000FD6 0000                         618+          DC    AL2(0) RESPONSE FIELD
000FD8 00                           619+          DC    AL1(0) TP-OP CODE
000FD9 00                           620+          DC    AL1(0) ERROR STATUS
000FDA 0000                         621+          DC    AL2(0) CSW STATUS
000FDC 00000000                     622+          DC    AL4(0) CURRENT ADDR LIST PTR
000FE0 00000000                     623+          DC    AL4(0) CURRENT ADDR POLL PTR
                                    624           END
```

POS.ID	REL.ID	FLAGS	ADDRESS
01	01	08	000019
01	01	08	0002B9
01	01	08	000F69
01	01	0C	000FC4

SYMBOL	LEN	VALUE	DEFN	REFERENCES
ABNORMAL	00001	0002C8	00321	0089 0124 0243 0247 0249 0251 0253 0255 0268 0272 0280 0284 0288
BASEREG	00001	00000B	00030	0066 0067 0068
BASEREG2	00001	00000C	00031	0067 0068 0069
CHKEOT	00001	000262	00278	0285
CHKIT	00001	0000A2	00118	0125
CHKTP20	00001	00027A	00286	
CHK41CC	00001	00023C	00266	0264
CHK7FCC	00001	000232	00262	0302
CKTP11	00001	00026E	00282	0277
CKTP20	00001	00025A	00275	0270
CLEAR	00001	00006D	00057	0164 0212
CLEARINT	00001	00017C	00196	0165 0213
CLOSE	00001	0002B4	00304	0193 0245
CLOSEMG	00001	000585	00406	0184 0411
CLOSEMGL	00001	00003C	00411	0185
CUDVSAVE	00002	000745	00558	0113
DCBR	00004	000F58	00572	0077 0079 0293 0299 0309 0430 0436 0448 0456 0468 0474 0487 0495 0507 0512 0613
DECBADDR	00001	000F68	00565	
DECBD	00004	000FBC	00609	0114 0259 0263 0267 0269 0271 0276 0279 0283 0287 0290 0426 0443 0463 0481 0502 0566
DSPTAB	00032	000720	00555	0090 0135
DSPTABRG	00001	000009	00028	0090 0135 0136 0149 0203
EIGHT	00001	000008	00045	0202
EIGHT8	00004	00071C	00554	0239
ENTER	00001	00007D	00059	0158 0206
ENTERINT	00001	000134	00168	0159
EOTRCVD	00001	000040	00054	
EOTRSPTX	00001	000040	00053	0279
ERALUNP	00001	0005C5	00418	0171 0420
ERALUNPL	00001	000004	00420	0172
FIVE	00001	000005	00041	0100 0191
FMTREG	00001	000008	00027	0134 0134 0136 0137 0148 0149 0202 0203
FMT0	00001	0000F0	00144	0140
FMT0SZ	00001	0000A7	00356	0096
FMT01	00001	0000F0	00147	0207
FMT1	00001	000110	00157	0141
FMT1SZ	00001	0000DA	00383	0151
FMT2	00001	00019C	00205	0142
FMT2SZ	00001	0000F7	00397	0198
FNDSEL	00001	0000C0	00126	0120
FORMATBR	00001	0000E4	00139	0137
FORMAT0	00001	0002E0	00335	0095 0356
FORMAT1	00001	000387	00358	0150 0383
FORMAT2	00001	000461	00385	0197 0397
FOUR	00001	000004	00040	0148 0158 0160 0162 0164 0206 0208 0210 0212
FOURTY1	00001	000041	00055	0267
H4096	00002	00071A	00553	0069
IHB0003A	00002	00003C	00087	0083
INAREA	00004	000750	00561	0107 0108 0108 0109 0109 0113 0119 0158 0160 0162 0164 0206 0208 0210 0212 0295 0450 0470 0489 0562
INAREAL	00001	0007D0	00562	
INDEX	00001	000744	00557	0127
INITIAL	00001	00004A	00094	0104
LAST	00001	000080	00061	0101 0123 0192

3/22/72

SYMBOL	LEN	VALUE	DEFN	REFERENCES
LNKREG	00001	00000A	00029	0097 0098 0099 0110 0111 0112 0152 0153 0154 0173 0174 0175 0188 0189 0190 0199 0200 0201 0218 0219 0220 0237 0239 0240 0265 0273 0274 0439 0459 0477 0498 0515
MSGADDR	00001	000004	00024	0095 0107 0150 0171 0184 0197 0216 0432 0491 0509
MSGLEN	00001	000005	00025	0096 0151 0172 0185 0198 0217 0429 0486
NOPRINT	00001	0001C0	00215	0178
NOPTR	00001	000558	00399	0216 0404
NOPTRL	00001	00002D	00404	0217
ONCEMORE	00001	00015C	00187	0195
ONE	00001	000001	00037	0103 0119 0121 0194
PA1	00001	00006C	00056	0160 0208
PA1INT	00001	00014C	00177	0161 0209
PA2	00001	00006E	00058	0162 0210
PA2INT	00001	000150	00179	0163
PA2INT1	00001	000150	00183	0211
POLDSPLY	00001	00070B	00543	0452
PTRTAB	00001	00000D	00032	
READ	00001	000072	00106	0102 0155 0166 0176 0204 0214 0221 0281
READBUF	00001	0005C1	00414	0416
READBUFL	00001	000004	00416	
READTI	00001	000600	00441	0110
READTRV	00001	000640	00461	
READTT	00001	0006BC	00500	
REGZERO	00001	000000	00021	
REG2	00001	000002	00022	0117 0117 0121 0121 0127 0128 0129 0130 0132 0136 0149 0203
RETCODE	00001	0001D8	00223	0098 0111 0153 0174 0189 0200 0219
RTNCDC	00001	00020C	00242	0229
RTNCDRG	00001	00000F	00033	0224
RTNCDTAB	00001	0001DC	00225	0224
RTNCD0	00001	000204	00236	0226
RTNCD1C	00001	00021C	00250	0233
RTNCD10	00001	000210	00244	0230
RTNCD14	00001	000214	00246	0231
RTNCD18	00001	000218	00248	0232
RTNCD20	00001	000220	00252	0234
RTNCD24	00001	000224	00254	0235
RTNCD4	00001	000206	00238	0227
RTNCD8	00001	00020C	00241	0228
SAMP327R	00001	000000	00001	
SAVE	00004	000F20	00563	0070 0071 0311 0330
SELDSPLY	00001	0006F0	00522	0091 0131 0186
SELREG	00001	000006	00026	0091 0100 0100 0101 0103 0103 0116 0119 0122 0122 0123 0131 0132 0133 0186 0191 0191 0192 0194 0194 0433 0492
SELSAVE	00004	000748	00559	0133
SEVEN	00001	000007	00044	
SEVENF	00001	00007F	00060	0263
SIX	00001	000006	00042	0122
SIXL	00004	00074C	00560	0129
SIXTN	00001	000010	00047	0271
SPECPOL	00001	0006FC	00533	0116
SSCHECK	00001	0002C8	00316	0115
SSMSG	00001	000010	00052	
START	00004	000042	00090	0080
THREE	00001	000003	00039	
TIMEOUT	00001	000001	00051	0271

3/22/72

SYMBOL	LEN	VALUE	DEFN	REFERENCES						
TP06	00001	000006	00043	0269						
TP11	00001	000011	00046	0283						
TP20	00001	000020	00048	0276	0287					
TWELVE	00004	000740	00556	0273						
TWENTY4	00001	000018	00049	0114	0279					
TWENTY8	00001	00001C	00050	0269	0276	0283	0287			
TWO	00001	000002	00038	0113	0113	0119	0119			
WAITD	00001	000228	00257	0099	0112	0154	0175	0190	0201	0220
WORKREG	00001	000003	00023	0128	0130					
WRITETI	00001	0005CC	00424	0097	0152	0173	0188	0199	0218	
WRITETIV	00001	000678	00479							
ZERO	00001	000000	00036	0101	0123	0136	0149	0192	0203	

```
NO STATEMENTS FLAGGED IN THIS ASSEMBLY
*STATISTICS*      SOURCE RECORDS (SYSIN) =    434       SOURCE RECORDS (SYSLIB) =  4875
*OPTIONS IN EFFECT*    LIST, DECK, NOLOAD, NORENT, XREF, NOTEST, ALGN, OS, NOTERM, LINECNT =  55
  784 PRINTED LINES
```

Where more than one page reference is given, the major reference is first.

INDEX OF BTAM AND SYSTEM MACRO INSTRUCTIONS

This index lists only the main discussions of the macro instructions.

GC30-2004-6

IBM System/360 Operating System BTAM (S360-30) Printed in U.S.A. GC30-2004-6

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

IBM System/360 Operating System
Basic Telecommunications Access Method

GC30-2004-6

READER'S COMMENT FORM

Your views about this publication may help improve its usefulness; this form will be sent to the author's department for appropriate action. Using this form to request system assistance or additional publications will delay response, however. *For more direct handling of such requests, please contact your IBM representative or the IBM Branch Office serving your locality.*

How did you use this publication?

- ☐ As an introduction
- ☐ As a text (student)
- ☐ As a reference manual
- ☐ As a text (instructor)
- ☐ For another purpose (explain)____

____

Please comment on the general usefulness of the book; suggest additions, deletions, and clarifications; list specific errors and omissions (give page numbers):

Cut or Fold Along Line

What is your occupation?____

Number of latest Technical Newsletter (if any) concerning this publication:________________________

Please include your name and address in the space below if you wish a reply.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

GC30-2004-6

Your comments, please . . .

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. Your comments on the other side of this form will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Cut or Fold Along Line

Fold Fold

First Class
Permit 40
Armonk
New York

Business Reply Mail

No postage stamp necessary if mailed in the U.S.A.

Postage will be paid by:

International Business Machines Corporation
Department 636
Neighborhood Road
Kingston, New York 12401

Fold Fold

IBM®

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)