GC30-3002-7
File No. S360/S370-30

Systems

IBM 3704 and 3705 Emulation Program Generation and Utilities Guide and Reference Manual

IBM

Eighth Edition (May, 1976)

This is a major revision of, and renders obsolete, GC30-3002-6. This edition applies to:

Emulation Program (OS) (Program No. 360H-TX-035) Version 3 Modification 0

Emulation Program (DOS) (Program No. 360H-TX-036) Version 2 Modification 3

and to all subsequent versions and modifications until otherwise indicated in new editions or Technical Newsletters.

Changes are continually made to the information herein; before using this publication in connection with the operation of IBM systems, consult the *IBM System/360 Bibliography* (GC20-0360) or *IBM System/370 Bibliography* (GC20-0001) and associated Technical Newsletters for the editions that are applicable and current.

Changes are indicated by a revision bar to the left of the change. Refer to the Summary of Amendments page for a description of the major additions and changes in this edition.

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be sent to IBM System Communications Division, Publications Center, Department E01, P.O. Box 12195, Research Triangle Park, North Carolina 27709. Comments become the property of IBM.

Preface

This publication provides the information necessary to define and generate an emulation program for the IBM 3704 and 3705 Communications Controllers, to load the program into the controller, and to dump the contents of the controller storage using the dump utility or the dynamic dump utility. The publication is directed to system analysts, system programmers, systems engineers, and field engineers responsible for defining or maintaining an emulation program to be used in communicating with a CPU in which TCAM, BTAM, QTAM (or an equivalent access method that supports IBM 2701, 2702, or 2703 transmission control units) is being executed.

The emulation program allows a 3704 and 3705 to perform most of the functions of an IBM 2701 Data Adapter Unit, an IBM 2702 Transmission Control unit, and an IBM 2703 Transmission Control unit, or any combination of the three.

Important: This publication applies only to (1) version 3 modification 0 (and subsequent versions) of the emulation program for OS (Program Number 360H-TX-035) and (2) version 2 modification 3 of the emulation program for DOS (Program Number 360H-TX-036). This publication does not apply to the emulation program/VS for OS/VS (Program Number 5744-AN1) and DOS/VS (Program Number 5744-AG1); the publication for these programs is *IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual* (GC30-3008).

This publication also contains the information needed by users of the IBM Airlines Control Program to specify communication lines on which airlines line control is used. (RPQ numbers 858911 and 858912 must be installed in the 3705 to which such lines are attached.) See Appendix J for information on how to specify communication lines requiring airlines line control (ALC).

This publication is arranged as follows:

Chapter 1 introduces the emulation program and summarizes the contents of the publication.

Chapter 2 describes the teleprocessing characteristics and generation options that the user must consider when defining an emulation program.

Chapter 3 describes the macro instructions used to define the emulation program.

Chapter 4 describes the emulation program generation procedure under the control of the Operating System (OS).

Chapter 5 explains how to use the loader utility, the dump utility and the dynamic dump utility programs respectively under the control of the Operating System.

Chapter 6 describes the emulation program generation procedure under the control of the Disk Operating System (DOS).

Chapter 7 explains how to use the loader utility, the dump utility and the dynamic dump utility programs respectively under the control of the Disk Operating System.

Appendix A lists the teleprocessing devices with which the communications controller can communicate when executing an emulation program.

Appendix B is an example of how to code an emulation program configuration under the control of the Operating System.

Appendix C gives the emulation generation and utility (loader, dump, dynamic dump) storage requirements for the Operating System and the Disk Operating System.

Appendix D lists the diagnostic messages produced by the emulation program generation process.

Appendix E and Appendix F list the diagnostic messages produced by the loader, dump, and dynamic dump utility programs for OS and DOS respectively.

Appendix G gives the character set and transmission code bit patterns for two codes used by World Trade Teletypewriter Terminals.

Appendix H contains a procedure for determining the interrupt priorities to be assigned to each line in the network.

Appendix I provides information on the use of the upper scan limit, address substitution, and high speed select options as specified in the GENEND macro.

Appendix J is intended for users who need to specify communication lines using airlines line control (ALC). The information in this appendix is of interest only to users of the IBM Airlines Control Program (ACP) and is applicable only when defining an emulation program for a 3705 in which RPQ numbers 858911 and 858912 are installed.

How to Use This Book

When first using this book, read Chapter 1 for a description of the purposes of the emulation program, the generation procedure, and the utility programs. Then read Chapter 2 for a description of the teleprocessing characteristics and generation options that you must consider when defining an emulation program.

After you have (1) identified the appropriate network configuration parameters, (2) recorded the details of your configuration in the *Installation Record*, and (3) become familiar with Chapter 2 in this manual, use the detailed macro instructions provided in Chapter 3 to define an emulation program that meets the needs of your teleprocessing installation. The coding conventions used in the macro instruction descriptions appear at the beginning of the chapter. Before coding the macro instructions, be sure that you are thoroughly familiar with and understand the operand descriptions given in this chapter.

For information on how to use the loader utility, the dump utility, or the optional dynamic dump utility, read chapter 5 (for OS) or chapter 7 (for DOS).

Refer to the appendixes for a coding example for the emulation program as well as for information on the types of stations supported, the storage requirements for the loader, dump, or the dynamic dump and for emulation program generation. In addition, diagnostic messages produced by the program generation procedure and the utilities are given.

Prerequisite and Related Manuals

Prerequisite to use of this publication is a basic understanding of teleprocessing and teleprocessing access methods. You should also have a general knowledge of the purposes of the IBM 3704 and 3705 Communications Controllers; this may be obtained from the publication, *Introduction to the IBM 3704 and 3705 Communications Controllers* (GA27-3051).

Contents

Figures

Summary of Amendments for GC30-3002-7

This eighth edition:

- Adds the following operands to the macros indicated:

  BUILD: CA, OPCSB2
  LINE: BUFSIZE

- Adds information on the IBM 3705-II. This edition applies equally to the 3705-I and 3705-II, except where a distinction between the two is explicitly made.
- Adds information on the type 3 communication scanner and type 4 channel adapter for the IBM 3705.
- Adds information on associating subchannel addresses with line addresses.
- Adds transparent ASCII to the codes that may be used on a BSC line serviced by a type 3 scanner.
- Adds Appendix G, which gives the character set and transmission code bit patterns for two transmission codes used by World Trade teletypewriter terminals.
- Adds Appendix H, which gives a method for determining the interrupt priority to be assigned to each line in the network.
- Adds Appendix I, which provides information on use of the upper scan limit, address substitution, and high speed select options as specified in the GENEND macro.
- Adds and updates messages in Appendixes E and F.

The foregoing changes, as well as other minor corrections and clarifications, are indicated by vertical lines to the left of the changed or added material.

Summary of Amendments for GC30-3002-6

This revision updates the publication as follows:

- The dynamic dump utility sections (OS and DOS) have been updated to support: (1) the new procedure for stopping the transfer of trace blocks to the host, (2) the new procedure for printing selected trace block entries, and (3) the PARM field option (specifying the number of lines to be printed per page) recognized by the dynamic dump utility (OS only).
- OS/VS and DOS/VS information for the emulation program has been removed from the manual. This information is now in the *IBM 3704 and 3705 Communications Controllers Network Control Program/VS Generation and Utilities Guide and Reference Manual (for OS/VS and DOS/VS VTAM users)* (GC30-3008).
- The list of emulation program modules that may be included during emulation program generation has been updated.
- The Unresolved External References list has been updated.
- Appendix F has been added and contains the diagnostic messages produced by the loader, dump, and dynamic dump utility programs running under the control of DOS. This information was previously in Appendix E which contained the utility diagnostics for OS and DOS. Appendix E now contains the OS diagnostic messages produced by the loader, dump, and dynamic dump utilities.
- Other clarifications and corrections appear in this edition where applicable.

Summary of Amendments for GC30-3002-5

The purpose of this edition is to incorporate TNL GN30-3023 into the base publication. This TNL revises the NEWSYNC operand and corrects the job control language in the examples in Chapters 4 and 5 as well as Figure 22 and 23 in Chapter 6. It also corrects the FEATURE operand specification in the example in Appendix B.

Chapter 1: Introduction

The IBM 3704 and 3705 Communications Controllers are programmed transmission control units that communicate with a variety of terminals, transmission control units, and computers. The programming can accommodate many different teleprocessing applications and operational requirements.

The emulation program resides in the 3704 and 3705 Communications Controllers and allows either controller to perform most of the functions of an IBM 2701 Data Adapter Unit, an IBM 2702 Transmission Control, an IBM 2703 Transmission Control, or as any combination of the three.

The emulation program is generated according to the network configuration of your installation. To begin the generation procedure you code a series of macro instruction source statements that *define* your teleprocessing network. The source statements are then *generated* by a compilation process into an emulation program load module. The load module is then *loaded* into the communications controller as your emulation program. You may obtain the contents of controller storage by *dumping* all or part of the communications controller. A dump allows you to examine the contents of storage as an aid to error diagnostics.

Defining, generating, loading, and dumping the emulation program are the subjects of this publication.

The Emulation Program

The emulation program (EP) allows many programs written for support of the 2701, 2702, and 2703 to operate with the communications controller with no modification. They include IBM Type I access methods that support the 2701, 2702, and 2703, as well as IBM Type II and Type III programs and user-written programs that interface with the 2701, 2702, and 2703 in a manner equivalent to IBM Type I access method programs. Programs that involve timing dependencies and support of certain special and custom features may have to be changed.

The emulation program requires that a type 1 or type 4 channel adapter be installed in the controller for attachment to a System/360 or System/370 byte-multiplexer channel. All models of the 3704 and the 3705 have enough storage to accommodate small emulation program configurations, but larger configurations require more than the minimum amount of storage.

The emulation program, in conjunction with the type 1 or type 4 channel adapter, permits the use of the same control sequences and data transfers as do the 2701, 2702, and 2703. It also provides most of the standard functions of these control units. Not supported are the parallel data adapter, synchronous data adapter type 1, the programmable two-processor switch, six-bit transcode, 230,400 bps synchronous speed, attachment to other than a byte multiplexer channel, direct attachment of the IBM 1032 Digital Time Unit, the IBM 2712 Remote Multiplexer attachment features, and the reverse channel feature. ASCII transparency is supported only for a communication line serviced by a type 3 communication scanner.

Appendix A contains a complete list of the types of stations supported by the communications controller when executing an emulation program.

Defining the Emulation Program

Achieving an operating emulation program is a three-step process. The first step, defining the program, is the most involved. Many different variables and options must be considered in preparing a program that meets the requirements of a particular network configuration and application.

An emulation program is defined in the form of a source program consisting entirely of EP generation macro instructions. These macro instructions include: (1) system macros which provide information pertaining to the entire controller such as hardware features, certain control program options, and program generation information such as data set names; (2) configuration macros which provide the information necessary to construct the tables needed by the emulation program to control the flow of data between the controller, the teleprocessing network, and the host processor; and (3) a generation delimiter macro that ends the program generation input stream. The source program, when preceded by the appropriate job control statements, forms the input to the next of the three steps, the generation procedure.

Chapter 2 describes each of the characteristics of the teleprocessing subsystem to be considered in defining an emulation program. Chapter 3 gives detailed descriptions of the macro instructions with which you define the emulation program.

Generating the Emulation Program

After the emulation program is defined in the form of a source program containing EP generation macro instructions, it is ready to be generated. The primary output of the generation procedure is an emulation program load module ready for loading into the communications controller.

Generating the Emulation Program Under OS

The emulation program generation procedure under the control of the Operating System (OS) is a two-stage process.

In the first stage of the generation procedure the EP generation macros you have coded are assembled by the communications controller assembler (IFKASM) or an OS assembler. The output from the assembler is a job stream containing the data and control statements necessary to create the desired emulation program. The job stream is a sequential data set that can be directed to cards, tape, or a direct access storage unit.

Operator intervention is required between the stages of program generation. Diagnostic messages produced at the end of stage one indicate any errors that have occurred. If these are serious errors, no job stream is produced. The source statements must be corrected and stage one must be re-executed. If no serious errors occur in stage one, the operator initiates the second stage, specifying as input the stage one output.

The second stage of the generation procedure assembles (using the controller assembler) the control tables that are required by the emulation program you are generating. Then the linkage editor is executed to combine the appropriate modules into an emulation program load module. This load module may then be loaded from the host processor into the communications controller.

Chapter 4 describes the emulation program generation procedure under the control of the Operating System.

Generating the Emulation Program under DOS

The emulation program generation procedure under the control of the Disk Operating System (DOS) is a two-stage process.

In the first stage of the generation procedure the macros you have coded are assembled by the communications controller assembler (IFTASM). The output from the assembler is an object module and a printed listing. The object module is placed (in stage two) in a relocatable library with existing object modules. The printed output lists, among other things, those modules (already in object form) that are required by the emulation program that you are generating. You must place these modules in a core image library in the link-edit step in stage two.

Operator intervention is required between the stages of program generation. Diagnostic messages are printed (stage one output listing) at the end of stage one indicating any errors that have occurred. If these are serious errors, no object module is produced. The source statements must be corrected and stage one must be re-executed. If no serious errors occur in stage one, the operator initiates the second stage.

Stage two of the generation procedure has two steps. In the first step, you use the MAINT utility program to catalog the object module produced by stage one in a relocatable library. In the second step, you link-edit the stage one object module with the pre-assembled emulation program load modules specified in the stage one output listing. This results in an emulation program load module that is located in a core image library. You must then use the CSERV utility program to move the load module to a file from where it can be loaded into the communications controller.

Chapter 6 describes the EP generation procedure under the control of the Disk Operating System.

The Utilities

The loader, dump, and dynamic dump programs help you start operation of the emulation program and locate errors that may occur during program execution. All three are utility programs, controlled by the appropriate job control and utility control statements and control cards. Each utility is described briefly below.

The Loader Utility

The final step in achieving an operating emulation program is that of loading the EP load module from the host processor into the communications controller. This is done with a loader utility program that is executed in the host processor, with the controller on line to the processor.

The loader utility is supplied as one of the EP system support programs.

Chapter 5 (for OS) and Chapter 7 (for DOS) explain how to use the loader utility.

Appendix E and Appendix F explain the diagnostic messages (OS and DOS, respectively) produced by the loader utility.

The Dump Utility

The dump utility program allows the entire contents of controller storage to be transferred from the controller to the host processor, which then prints all or part of the contents in hexadecimal form. You can request a formatted dump or an

unformatted dump of controller storage. The unformatted storage dump is printed with EBCDIC equivalents on the right side of the page. In a formatted storage dump, the mnemonic operation codes are printed with the instructions.

Executing the dump utility stops operation of the emulation program. After the dumping process is completed, an emulation program must be reloaded into the controller before teleprocessing operations can resume.

The dump utility, like the loader, is supplied as one of the EP system support programs.

Chapter 5 (for OS) and Chapter 7 (for DOS) explain how to use the dump utility program.

Appendix E and Appendix F explain the diagnostic messages (OS and DOS respectively) produced by the dump utility.

The Dynamic Dump Utility

The dynamic dump program is an optional utility program that allows the contents of controller storage to be transferred from the controller to the host processor without stopping the operation of the emulation program. A full storage dump or a dump of the trace table can be obtained. In addition, portions of storage can be displayed on the operator's console at the host processor. The utility can also activate or deactivate the EP line trace function and allows the selection of two program levels to be traced.

Chapter 5 (for OS) and Chapter 7 (for DOS) explain how to use the dynamic dump utility program.

Appendix E and Appendix F explain the diagnostic messages (OS and DOS, respectively) produced by the dynamic dump utility.

Chapter 2: Defining the Emulation Program

This chapter describes the many aspects of a teleprocessing subsystem that you must identify to the emulation program to meet the requirements of a particular network configuration and application.

The chapter is divided into six major sections. The first three explain the characteristics of a teleprocessing subsystem with respect to:

- The stations, lines, and modems of the teleprocessing network
- The communications controller hardware configuration
- The type of transmission control unit to be emulated

The remaining sections explain the optional service aid facilities that may be included in the program and the program generation options and data sets (files) that the generation procedure will use in creating the emulation program load module.

The description of each characteristic and option given in this chapter is not exhaustive; it is intended to provide sufficient information to enable you to select the appropriate parameters when coding the EP generation macro instructions given in Chapter 3.

Once you are familiar with those characteristics that apply to your equipment configuration and applications, you are ready to code the program generation macro instructions.

Teleprocessing Network Characteristics

The following descriptions of network characteristics give the names of the applicable macro instructions and operands. For your convenience, many operands of the LINE macro (representing one communication line attached to the communications controller) can be specified instead in the GROUP macro (representing a number of communications lines with certain common characteristics) as explained in Chapter 3. The description of the operand always appears under the macro named, however. The TERM operand, for example, may be coded on the GROUP macro, but the description of that operand is given under the LINE macro.

Station Characteristics

In this manual, *station* refers to any equipment, regardless of type, that can transmit data onto, or receive data from, a communication line connected to the communications controller. This definition includes (1) computers, (2) transmission control units such as the IBM 2701, 2702, and 2703, and (3) the input/output devices (keyboards, printers, tape and card readers, punches, and display screens) usually referred to as *terminals.*

Type of Station

Type of station means the numerical designation by which the station is known, or an abbreviation thereof (for example, 1050, 2780, SYS3 [System/3]). Appendix A lists the types of stations with which the communications controller can communicate when executing an emulation program. The type of station with which the controller communicates over a communications facility is specified in the TERM operand of the LINE macro (or, for certain types of stations, in the CUTYPE operand of that macro).

For some types of terminals and control units, the presence or absence of the features below is specified in the FEATURE operand of the LINE macro representing the line over which the controller communicates with the terminal.

Longitudinal Redundancy Checking: Longitudinal redundancy checking is a technique of record checking performed at the receiving station after a block check character has been accumulated. Some start-stop stations have a record checking capability and others do not. For each line in the network, you must specify to the emulation program whether the terminals with which the program communicates over that line have the record checking capability. If the terminal is an IBM 1050, 1060, 2260, 2265, 2845, 2848 or System/7, which do have this capability, specify LRC in the FEATURE operand of the LINE macro. Also specify LRC for an IBM 2740 (Model 1 or 2), if it is equipped with the Record Checking feature. For other types of start-stop terminals, specify NOLRC.

Downshifting on Space Characters: Some AT & T 83B3, Western Union 115A, and World Trade teletypewriter (teleprinter) terminals, upon sending or receiving a space character, automatically downshift so that subsequent message text is in lower-case, or down-shifted, mode. Automatic downshifting avoids the need to send a LTRS character to effect downshifting. In the FEATURE operand of the LINE macro, specify SPACE if the terminals are so equipped.

Immediate End: Upon receiving an end-of-transmission character from a start-stop terminal, the emulation program normally delays ending the receive operation for several character times (the time required for the transmission of several characters) until the line becomes electrically "quiet". The absence of further characters following the EOT verifies that the EOT character is valid and not a data character converted by line noise to a false EOT. Checking for false EOTs in this manner is appropriate for many applications. In some applications, however, the terminal continues to send data immediately after sending the EOT (as when the terminal is transmitting from a paper tape in which data interspersed with EOTs is punched). If the end of the receive operation were in this case delayed, the program would not recognize the EOT because of the immediately following data characters. In this instance it is necessary to specify IMEND in the FEATURE operand; this causes the program to end the receive operation immediately upon detecting the EOT, without waiting to detect the presence or absence of any following characters.

Dual Code: Either of two transmission codes (for example, EBCDIC or ASCII) can be transmitted on a binary synchronous communication line attached to an IBM 2701 Data Adapter Unit equipped with the Dual Code feature for that line. The code used is changed from one to the other by command from the access method. The same function can be performed when the IBM 3704 or 3705 is installed in place of the 2701. Specify DUALCODE in the FEATURE operand of the LINE macro representing the line if the Dual Code feature was used for that line when the line was attached to the 2701. Otherwise, specify NODUALCD or omit the parameter.

For a line serviced by a type 3 scanner, transparent ASCII (as well as ASCII and EBCDIC) can be one of the two codes used on the line.

The codes used on the line are specified in the CODE operand of the LINE macro.

Communication Line Characteristics

A *communication line* as used in this book includes the entire transmission link between a station and the communications controller, including the modems (data sets), regardless of the actual transmission medium—physical conductors (wire), microwave link, satellite link, etc., or a combination of these.

Line characteristics refer to (1) the functional attributes of the transmission path, for example, whether the line is half-duplex or duplex; (2) logical characteristics, such as the transmission code and line control scheme employed; and (3) related aspects of the line such as the line interface address or subchannel address by which it is known to the emulation program. The line interface addresses can be obtained from the *Configuration Data Sheet* available from your IBM representative prior to shipment of the communications controller.

Stations may communicate with the communications controller using any of three kinds of line connections: nonswitched point-to-point, nonswitched multipoint, and switched point-to-point. (Not all types of stations can communicate with the controller over all three kinds of line connections.) You must code a LINE macro for each line connected to the communications controller, regardless of the kind of line. This macro specifies to the emulation program some (but not all) of the characteristics of the line. Whether a line is switched or nonswitched must be specified in the DIAL operand of the GROUP macro representing the line group. However, you need not specify whether a nonswitched line is multipoint or point-to-point. It is the responsibility of the access method to be aware of this, and to issue the appropriate command sequences for each type of line.

Half-duplex vs. Duplex Lines

The emulation program must know whether a communication line is half-duplex or duplex (full-duplex). You specify this in the DUPLEX operand of the LINE macro. This operand represents the characteristics of the entire communications path including common-carrier lines and equipment, and the modems at both ends of the path. The operand does not specify the mode of data transfer over the line, which is always half duplex for any start-stop or BSC station with which the controller can communicate. (It is important not to assume that a two-wire modem is necessarily a half-duplex modem; some such modems are in fact duplex. In general, if the "clear-to-send" signal lead in the modem is continuously activated, the modem is duplex, regardless of whether it is a two-wire or four-wire modem. If you are in doubt, consult the supplier or installer of the modem.)

Line Speeds and Clocking

In the SPEED operand of each LINE macro, specify the data rate at which the communication line is to operate. This is the rate at which the station, controller, and modems are designed to transmit data over the communications facility that links the station and the controller.

If the modem that connects the line to the controller has two possible data rates, as is the case with the IBM 3872 and 3875 modems, for example, designate in the DATRATE operand of the LINE macro whether the line is to operate at the higher or lower of the two rates.

In the CLOCKNG operand of the LINE macro, specify whether internal (business-machine) clocking or external (modem) clocking is used for the communication line. Internal clocking is provided by the communication scanner to which the line is connected. External clocking is provided by the modem, whether the modem is a separate unit or built into the controller.

Each communication scanner in the communications controller may be provided with from one to four oscillators. The bit rate for each oscillator must be specified in the SPEED operand of the corresponding CSB macro.

Line and Subchannel Addresses

Each communication line attached to the controller requires a nonshared subchannel address on a byte-multiplexer channel. Each line is identified to the emulation program by a line interface address representing the physical location within the controller at which the line is attached (via line set and line interface bases [LIB]). Associated with each line address are one or more CPU subchannel addresses. The multi-subchannel line access (MSLA) facility of the emulation program permits two or more emulation subchannels to communicate, alternately, with the same communication line. When transferring message data to or from a line, the emulation program uses the line interface address to communicate with the station and uses the associated subchannel address to communicate with the host processor.

Specify the line interface address and the associated subchannel address in the ADDRESS operand of the corresponding LINE macro.

Native and Emulation Subchannel Addresses

The native subchannel is a dedicated subchannel of a byte-multiplexer channel. The host processor has access to the communications controller for initial program load (IPL), dump, and dynamic dump procedures through the use of this subchannel. The native subchannel address must be specified during operating system generation in the unit control block (UCB), for OS; or the physical unit block (PUB), for DOS.

The emulation program operates similarly to the 2703 Transmission Control in controlling its subchannels. The range of subchannel addresses to be recognized by the emulation program is specified in the LOCHAN and HICHAN operands of the BUILD macro.

Modems and Automatic Calling Units

The following information on modems and automatic calling units (ACU) attached to the communications controller must be specified to the emulation program.

New Sync Feature

Certain types of synchronous modems are equipped with a feature called "new sync", which reduces the amount of line turnaround time that is normally expended each time the direction of transmission on the line is reversed. The NEWSYNC operand of the LINE macro specifies whether this feature is to be used.

NEWSYNC=YES is valid only if the modem (at the controller) serving the line has the new sync feature, *and* if the communications controller is the multipoint master (not tributary) station for a duplex (not half-duplex) line on which multipoint line control is used.

Determine from your IBM representative or the installer or supplier of the modem (if other than an IBM modem) whether the modem has the new sync feature.

Ring Indicator Mode (not applicable in U.S. and Canada)

Certain European modems may require that their 'ring-indicator' signal line be energized (signifying that the modem is being called by a station) before the communications controller indicates its readiness to receive by energizing the modem's 'data terminal ready' signal line. (These and other signal lines constitute the interface between the communications controller and the modem.) If this requirement applies for a modem in your network, code RING=YES in the LINE macro for the communication line attached to the modem. Most modems do not have this requirement, and for these you would specify RING=NO in (or omit the RING operand from) the LINE macro. Specifying RING=YES for a modem that does not have this requirement can result in an unnecessary delay in establishing the connection.

Automatic Calling Units

Any switched call-out line controlled by the emulation program must be equipped with an automatic calling unit (ACU). Specify the line interface address to which the ACU is attached in the AUTO operand of the LINE macro representing the line for which the ACU is used.

Communications Controller Hardware Configuration

Some of the hardware options with which the communications controller is equipped must be identified to the emulation program. These options are: (1) the type(s) and number of channel adapters that join the controller to the host processor; (2) the type, number, and oscillator bit rates of the communication scanners installed in the controller; and (3) the interrupt priority to be used for each line serviced by a scanner.

In the CA operand of the BUILD macro, specify the type(s) of channel adapter installed in the controller.

The communications controller can be equipped with from one to four communication scanners. The IBM 3704 and the IBM 3705 models A1, A2, and E1–E8 always have a single scanner. Models B1–B4 of the 3705-I can have one or two scanners; models C1–C6 up to three scanners; and models D1–D8 up to four scanners. Models F1–F8 of the 3705-II have one or two scanners; models G1–G8 have three scanners; and models H1–H8 have four scanners. Each communication line attached to the controller is serviced by one of the scanners. The number of lines serviced by each scanner depends upon the data rates (line speeds) at which the lines operate. Each scanner may be equipped with from one to four oscillators, or internal clocks, and can therefore provide internal clocking for up to four different speeds of lines. In addition, the scanner may service lines for which external modems (including integrated modems within the 3704 or 3705) are used, without restriction as to the number of different external clock speeds used for those lines. To service a line that is externally clocked, however, a scanner must be equipped with an oscillator that operates at less than one-half of the data rate of that line. (This may be the same oscillator that furnishes clocking for one or more of the internally clocked lines.) A scanner equipped with 600 bps and 1200 bps oscillators, for example, could service lines operating at these speeds, using *internal* clocking, and also service lines using *external* clocking at speeds exceeding 1200 bps—for instance, 2000 and 7200 bps. The scanner could not, however, service externally clocked lines of 1200 bps or less, because in this example there is no oscillator that operates at less than one-half of 1200 bps.

For each scanner, you must specify to the emulation program (1) the type of scanner, (2) the machine module in which it is installed, and (3) the bit rates of

the oscillators with which it is equipped. This information should be obtained from the system designer before you code the program generation macro instructions. Specify the details of the scanners in the TYPE, MOD, and SPEED operands of a CSB macro—one macro for each scanner in the controller.

The emulation program is interrupted by the line interface hardware of the controller each time a data bit, a data character, or a data buffer (depending on the type of scanner) is sent over or received from a communication line. To avoid bit or character overrun or underrun, lines having a high data rate require service from the program more frequently than lines having lower data rates. Each line serviced by a given communication scanner is therefore assigned an interrupt priority relative to other lines serviced by the same scanner. If all lines on the scanner have the same data rates, the priorities may be equal. If the lines have differing rates, however, those with high rates should be assigned higher priority than those with lower rates.

For a type 1 scanner, the priority may be 0 or 1 (1 is the higher priority). For a type 2 or type 3 scanner, the priority may be 0, 1, 2, or 3 (3 is the highest priority). The priority values are specified in the INTPRI operand of the LINE macro.

Appendix H describes a method for determining the interrupt priority for each line in the network.

Associated with each communication line serviced by a type 3 scanner is a pair of buffers contained within the control blocks related to the line. The size of each buffer in the pair is user specified as 4, 8, 16, 32, 64, 96, 128, 160, 192, or 224 bytes. For a given amount of data passing between host processor and communication line, use of larger buffers results in fewer channel and scanner service interrupts than does use of smaller buffers. The consequent reduction in interrupt-processing overhead minimizes the possibility of overruns which can result from temporary slowdowns in channel operation or temporary peaks in data traffic over the network.

The size of the emulation mode buffers for a line serviced by a type 3 scanner is specified in the BUFSIZE operand of the LINE macro for the line. If you do not specify a size, 32-byte buffers are provided for lines operating at speeds of 19,200 bps or less, and 64-byte buffers are provided for lines operating at higher speeds (as specified in the SPEED operand of the LINE macros).

Transfer of data between the host processor and the line occurs in a manner equivalent to that provided by the IBM 2701, 2702, or 2703 being emulated. In the CU operand of the LINE macro, specify the type of transmission control unit to be emulated for the line—2701, 2702, or 2703.

Procedural Options

When the emulation program operates a communication line, transfer of data between the CPU and the line occurs in a manner similar to that provided by the IBM 2701, 2702, or 2703 being emulated. In the CU operand of the LINE macro, you specify the transmission control unit—2701, 2702, or 2703— being emulated for that line.

When defining an emulation program, there are three important operational characteristics to be specified relating to: the type of line control discipline to be used for each line, the terminal time-outs required, and—for World Trade tele-

typewriters only—the end-of-block and end-of-transmission sequences to be recognized by the program.

Type of Line Control

All types of stations with which the communications controller can communicate under emulation program control use one of two line control disciplines: binary synchronous (BSC) and start-stop (or asynchronous). Each communication line attached to the controller uses either BSC or start-stop line control; the same line never uses both types.

The type of line control discipline used is specified in the LNCTL operand of the GROUP macro. (All lines in a group must use the same line control discipline.)

Terminal Time-outs

For each communication line the emulation program normally observes two *time-out* intervals of several seconds' duration. One of these intervals is the *reply time-out*, which limits the amount of time the program will await a station's response to polling or response to message data sent to the station. The other interval is the *text time-out*, which limits the time that may elapse between receipt of successive message characters from the station after message transmission has begun. If the time-out expires before the response or the next message character is received, the program ends the read operation for that station and notifies the access method of a time-out error. These time-outs apply to each line in the network.

By observing these two time-out intervals, the emulation program prevents a communication line from being idled indefinitely because of excessive delay in entering successive message characters at a station or because a malfunction or power failure at the station interrupts its transmission to the communications controller.

Unless you specify different values in the REPLYTO and TEXTTO operands of the GROUP macro, the emulation program uses the default time-out intervals indicated in the descriptions of these two operands for all lines in the group represented by that macro.

Some applications may justify unlimited intervals, that is, no time-out at all. This may also be specified in the REPLYTO or TEXTTO operands.

EOB and EOT Sequences for World Trade Teletypewriter Terminals

You may specify the character sequence the emulation program is to recognize, when receiving from a terminal, as the end-of-block (EOB) and end-of-transmission (EOT) sequences.

The EOB sequence may be either FIGS *x* or *nnnn*. *x* and *n* may be any telegraph code combination except a combination representing the FIGS or LTRS character. (If the terminal is equipped to send who-are-you (WRU) sequences, *x* also may not be the letter D.)

The EOT sequence may be FIGS *y* LTRS; *y* may be any applicable telegraph code combination except one representing FIGS, LTRS, or the same *x* character used in the EOB sequence, FIGS *x*.

Specify the required sequences in the EOB and EOT operands of the GROUP macro representing a World Trade teletypewriter (teleprinter) line group.

Note: Appendix G lists the character sets and transmission code bit patterns for the ITA2 and ZSC3 codes.

Multi-Subchannel Line Access Facility

The multi-subchannel line access (MSLA) facility of the emulation program allows the program to communicate over two type 4 channel adapters concurrently. The channel adapters may both be attached to the same host processor or may be attached to separate processors. The MSLA facility further allows two or more CPU subchannels (on the same or different channels) to communicate, alternately, with the same communication line. In operation, a command issued over one of the subchannels seizes the line for use of that subchannel and the access method using that subchannel. The access method retains use of the line via that subchannel until it issues a Disable command, thus releasing the line and thereby freeing it for use by another subchannel. (Alternatively, the 3705 control panel can be used to release a line from control of one subchannel in order to switch it to another subchannel. This action is required if the access method using the line does not issue operator-controlled Disable commands.)

Subchannel-to-line associations are established during program definition and can be changed only by respecifying the associations and regenerating the program.

The physical characteristics of the line (such as type of line control, line speed, etc.) remain constant regardless of which subchannel is currently using the line. The use of the line by each subchannel must be consistent with the line characteristics. Violation of this requirement will cause unpredictable results when the access method communicates with the line.

The MSLA facility can be used in the following ways:

- Load balancing—communication lines can be switched from one host processor to the other during high-traffic periods to balance the load on the processors.
- CPU backup—communication lines can be switched to a backup host processor if the original host processor, channel, or access method fails. Execution of the control program does not end, and the program need not be reloaded into the communications controller.
- Line sharing—two access methods in the same or different host processors can share the same communication line, alternately. The same line can thus be assigned to different applications at different times of day.

The description of the ADDRESS operand of the LINE macro explains how to associate subchannels with a line.

Diagnostic and Service Aids

The emulation program provides the line trace, panel test, and dynamic dump facilities to aid in diagnosing difficulties in network operation. Although optional, inclusion of these facilities in the program is recommended.

Line Trace Facility

The line trace facility of the emulation program is a service aid that permits detailed analysis of the operation of any communication line controlled by the program. This facility records operating parameters of a line each time a level two (except bit service) or level three interrupt occurs for that line. (Level two is the program level at which bit service or character service for the communication line is performed. Level three is the program level at which servicing of channel interrupts is performed.) The program accumulates this information in a trace table within controller storage. The contents of the controller storage must be

dumped (using the dump or the dynamic dump utility) in order to make the line trace records available. Inclusion of the line trace facility, the number of lines to be traced, and the size of the trace table are specified in the LINETRC operand of the BUILD macro.

The line trace facility does not interfere with normal operation of the communication line. Performance may diminish somewhat because of the additional processing needed each time a character service interrupt occurs for the line or lines being traced. The amount of decrease in performance depends upon how heavily the communications controller is currently loaded. Inclusion of the line trace facility has no effect on performance except when a line is actually being traced.

Line traces are initiated at the control panel of the communications controller. Any number of lines may be traced concurrently. Refer to the *Control Panel Guide* for the procedure for initiating the line trace facility from the control panel.

Panel Tests

Certain tests of communication lines can be run from the control panel of the communications controller. These optional tests, called panel-initiated line tests (or panel tests), are explained in the *Control Panel Guide.* Using the test routines, the operator at the controller can perform many of the teleprocessing functions (such as polling, addressing, and data transfer) normally executed by the controller and its control program upon request from the access method.

The panel-initiated line tests can be run only if the test function is included in the emulation program by specifying TEST=YES in the BUILD macro.

Refer to the *Guide to Using the IBM 3705* (or *3704*) *Communications Controller Control Panel* for the control panel procedure that explains how to run the panel test (line test).

Dynamic Dump Facility

The dynamic dump facility is a service aid that transmits communications controller storage contents to the host processor without stopping execution of the emulation program. A full storage dump or a dump of the trace table can be obtained. Additionally, the line trace can be activated or deactivated from the control panel. Portions of controller storage can also be displayed on the operator's console at the host processor.

The DYNADMP operand of the BUILD macro specifies whether the dynamic dump option is to be included in the emulation program and specifies the emulation subchannel address(es) over which the controller storage is to be dumped. Each channel adapter in the controller can have one CPU subchannel address assigned for this purpose; the assigned subchannel(s) cannot be used for communicating with any line in the network.

Program Generation Options and Data Sets (Files)

All of the options described thus far in this chapter have related to the operational characteristics of the teleprocessing subsystem. Described in this remaining section are several options affecting the generation procedure and the emulation program data sets (files) used in the procedure.

Program Generation Options

Program generation options pertain to the type of functions that the emulation program is to perform, the type of communications controller in which the pro-

gram will be executed (3704 or 3705), and several assembly and link-editing options. All program generation options are specified in the BUILD macro.

Model of Controller

The same emulation program can be executed in an IBM 3704, 3705-I, or 3705-II Communications Controller. However, minor internal differences between the three controllers require that you specify, in the MODEL operand of the BUILD macro, the type of controller in which the emulation program is to be loaded and executed. Changing the value in this operand is the only modification required to allow an emulation program originally defined for one type of controller to be executed in the other type, *provided* that the subsystem configurations are identical. That is, the network configuration (including line and subchannel address assignments), the controller configuration (number and type of channel adapters and communication scanner), and procedural options must be the same for both controllers.

Other Options

The remaining program generation options, and the operands of the BUILD macro by which you specify them are: (1) Whether stage two of the generation procedure is to consist of a single, multi-step job or a separate job for each step, and whether a job card is required (JOBCARD); (2) the region size for the stage two linkage editor job steps (LESIZE); and (3) the type of device or class of devices to be used for utility data sets during stage two (UNIT).

These options are applicable only when you are generating an emulation program under OS; they are not applicable for DOS.

Data Sets (Files) Used in the Generation Procedure

The names of the various program data sets (files) to be used in the generation procedure are specified by the LOADLIB, OBJLIB, QUALIFY, UT1, UT2, and UT3 operands of the BUILD macro.

The NEWNAME operand of the BUILD macro specifies the name to be given to the generated program load module.

These data set (file) names are applicable only when you are generating an emulation program under OS; they are not applicable for DOS.

Name	*Operation*	*Operands*
[symbol]	BUILD	HICHAN=(addr1[,addr2]), LOADLIB=dsname, (OS only) LOCHAN=(addr1[,addr2]), OBJLIB=dsname, (OS only) [,CA=(adapter1[,adapter2])] (OS only) [DYNADMP={{YES,[{NSC }][,{addr1} {NSC }]}{addr2}] {NONE } {NONE } } {NO } [,JOBCARD={YES}] (OS only) {NO } [,LESIZE=size] (OS only) [,LINETRC=({YES}[,lines])] {NO } [,MODEL={3705-2}] {3705 } {3704 } [,NEWNAME={EP001 }] (OS only) {symbol} [,OPCSB2={YES}] {NO } [,QUALIFY={symbol}] {NONE } {SYS1 } [,TEST={YES}] {NO } [,TYPSYS={OS }] {DOS} [,UNIT=unit type] (OS only) [,UT1=dsname] (OS only) [,UT2=dsname] (OS only) [,UT3=dsname] (OS only)

[symbol]

Provides a name for the macro. *symbol* may be any valid one to eight character assembler language symbol. The generation procedure does not check the symbol for validity.

Chapter 3: Emulation Program Generation Macro Instructions

This chapter gives detailed descriptions of the macro instructions with which you define an emulation program.

Macro Instruction Coding Conventions

The following conventions are used in the descriptions of the emulation program macro instructions.

- Capital letters represent values that you code directly, without change.
- Small letters represent parameters for which you supply a value.
- Brackets [ ] enclose operands or symbols that are either optional or conditional.

  An optional operand is one that you may choose to code or omit, independent of other operands you may code or omit. Depending on the operand, omitting it may cause emulation program coding for the corresponding feature or function to be omitted or included, or omitting it may cause a specific numeric value (default value) to be given. The assumed value is always given.

A conditional operand is one that you may need to code or omit, depending on how you code (or omit) other operands in the same macro or a different one. For each conditional operand, the conditions under which you should code or omit it are indicated.

- Parentheses ( ) must be used to enclose a sequence of values coded on one operand.
- Braces { } indicate that an operand has a value which you must choose from the enclosed items.
- An ellipsis ... indicates that you may code a sequence of values, within parentheses.
- An underlined value represents the default value of the operand; that is, the emulation program will use that value if you omit the operand.
- Quotes must be used to frame a character string if it can be confused with a keyword value for an operand. This is to avoid preventing your use of certain names as symbols.

Symbols coded in the name field of a macro instruction must not begin with a $ character.

System Definition Macro Instruction

This section contains the single system definition macro instruction, BUILD, to be used in defining an emulation program.

BUILD Macro Instruction

The first macro instruction in the program source statements is BUILD. This macro specifies such information as:

- The type of program to be generated, namely, a program that emulates the operation of transmission control units (IBM 2701, 2702, and 2703).
- The type of operating system (OS or DOS) that is to control the operation of the emulation program.
- The type of controller (3704, 3705-I, or 3705-II) that is to execute the emulation program.
- The name that is to be assigned to the EP load module.
- The range of CPU subchannel addresses to be used by the emulation program.
- Certain optional facilities that may be included in the program.
- Certain program generation options that may be desired.
- The names of the program data sets used in the program generation process.

HICHAN=(addr1[,addr2])

Specifies the upper end of the range of subchannel addresses to be associated with the channel adapters(s) installed in the communications controller, as follows.

If the controller has a single type 1 or type 4 channel adapter, specify HICHAN=(addr1) (parentheses may be omitted). *Example:* HICHAN=2B. If the controller has two type 4 adapters, specify HICHAN=(addr1,addr2), where *addr1* is the highest subchannel address associated with the type 4 channel adapter in the *base* module of the 3705, and *addr2* is the highest subchannel address associated with the type 4 adapter in the first *expansion* module. The value of *addr1* and *addr2* must be one of the hexadecimal addresses shown below.

This operand defines the highest subchannel address on each channel adapter to be associated with any line (or the address of the subchannel used for the dynamic dump operation) on that channel adapter. The address you specify must therefore equal or exceed the highest emulation subchannel address specified in the ADDRESS operand of any LINE macro (or the address specified in the DYNADMP operand of the BUILD macro).

The range of subchannel addresses specified by HICHAN and LOCHAN must not include any addresses associated with shared UCWs (unit control words) in the host processor.

For a program to be executed in a 3705, the address must equal $4n-1$, where n is an integer equal to or exceeding 1.

Choose the appropriate subchannel address from the list below:

03	07	0B	0F	83	87	8B	8F
13	17	1B	1F	93	97	9B	9F
23	27	2B	2F	A3	A7	AB	AF
33	37	3B	3F	B3	B7	BB	BF
43	47	4B	4F	C3	C7	CB	CF
53	57	5B	5F	D3	D7	DB	DF
63	67	6B	6F	E3	E7	EB	EF
73	77	7B	7F	F3	F7	FB	FF

Note: Specifying an address that is not listed causes an MNOTE warning message to appear in the assembly listing.

For a program to be executed in a 3704, any *odd* subchannel address between 01 and FF, inclusive, is valid. (Ignore any MNOTE warning message that indicates that an invalid address is specified; this message applies only to a program to be executed in a 3705.)

This operand is required.

Note: Also see description of the LOCHAN operand.

LOADLIB=dsname *(generation under OS only)*

Specifies the name of a partitioned OS data set that will contain the load module produced by the program generation procedure. The data set name may be simple or qualified, as determined by the QUALIFY operand of this macro. The simple data set may be up to eight characters long; the first must be alphabetic or $, #, or @. This data set must be cataloged.

This operand is required for generation under OS.

LOCHAN=(addr1[,addr2])

Specifies the lower end of the range of subchannel addresses to be associated with the channel adapter(s) installed in the communications controller, as follows.

If the controller has a single type 1 or type 4 channel adapter, specify LOCHAN=(addr1) (parentheses may be omitted). *Example:* LOCHAN=20. If the controller has two type 4 adapters, specify LOCHAN=(addr1,addr2), where *addr1* is the *lowest* subchannel address associated with the type 4 channel adapter in the *base* module of the 3705, and *addr2* in the lowest subchannel address associated with the type 4 adapter in the first *expansion* module. The value of *addr1* and *addr2* must be one of the hexadecimal addresses shown below.

This operand defines the lowest subchannel address on each channel adapter to be associated with any line (or the address of the subchannel used for the dynamic dump operation) on that channel adapter. The address you specify must therefore be less than or equal to the lowest emulation subchannel address specified in the ADDRESS operand of any LINE macro (or the address specified in the DYNADMP operand of the BUILD macro).

The range of subchannel addresses specified by HICHAN and LOCHAN must not include any addresses associated with shared UCWs (unit control words) in the host processor.

For a program to be executed in a 3705, the address must equal $16n$, where n is an integer equal to or exceeding 0. Choose the appropriate subchannel address from the list below:

00	10	20	30
40	50	60	70
80	90	A0	B0
C0	D0	E0	F0

Note: Specifying an address that is not listed causes an MNOTE warning message to appear in the assembly listing.

For a program to be executed in a 3704, any *even* subchannel address between 00 and FE, inclusive, is valid. (Ignore any MNOTE warning message that indicates that an invalid address is specified; this message applies only to a program to be executed in a 3705.)

Note: Optimum storage utilitization is achieved by a contiguous assignment of all emulation subchannels. Each unassigned subchannel address between the values specified by the LOCHAN and HICHAN operands adds ten bytes to the EP storage requirements.

Caution: (1) all commands (except Sense, Test I/O, and I/O No-op) issued to unassigned subchannels within the HICHAN-LOCHAN range will be rejected. (Unassigned means that the subchannel is not specified in the ADDRESS operand of any LINE macro or in the DYNADMP operand of the BUILD macro.) (2) Although the channel adapter recognizes as valid any commands issued for a subchannel address that is outside the HICHAN-LOCHAN range, the emulation program does not recognize the address and therefore ignores any such commands received from the CPU channel. A permanently busy ("hung") subchannel results. (3) If a unit control block (UCB) exists for a device associated with a subchannel outside the HICHAN-LOCHAN range, but within the channel adapter's address range, initial program load (IPL) of the operating system in the host processor cannot be completed because Test I/O and Sense commands—though accepted by the controller—are ignored.

The range specified by the LOCHAN and HICHAN operands applies only to emulation subchannels.

The native subchannel address (the load or IPL subchannel address) need not fall within this range.

This operand is required.

`OBJLIB=dsname` *(generation under OS only)*

Specifies the name of a partitioned OS data set that will contain the output from all assemblies during stage two of the generation procedure. The data set name may be simple or qualified, as determined by the QUALIFY operand of this macro. The simple data set name may be up to eight characters long; the first must be alphabetic or $, #, or @. This data set must be cataloged.

This operand is required for generation under OS.

`[CA=(adapter1[,adapter2])]`

Specifies the type(s) of channel adapter installed in the communications controller.

`adapter1`

Specifies the type of adapter in the base module of the 3705 or in the 3704.

`adapter2`

Specifies the type of adapter in the first expansion module of the 3705 (not applicable for a 3704).

See Figure 3-1 for valid specifications of the CA operand.

Adapter Configuration		*Adapter Specification*
First CA (base module)	*Second CA (expansion module)*	*CA=*
Type 1	(none)	TYPE1[3]
Type 4	(none)	TYPE4[4]
Type 1	Type 2	(TYPE1,TYPE2)[1]
Type 1	Type 3	(TYPE1,TYPE3)[2]
Type 4	Type 2	(TYPE4,TYPE2)[1]
Type 4	Type 3	(TYPE4,TYPE3)[2]
Type 4	Type 4	(TYPE4,TYPE4)

[1]Valid only if Type 2 CA is disabled.
[2]Valid only if Type 3 CA is disabled.
(Type 2 and Type 3 channel adapters are not supported by the emulation program.)

If CA operand is omitted:

[3]CA=TYPE1 is assumed if MODEL=3704 or 3705 is specified.
[4]CA=TYPE4 is assumed if MODEL=3705-2 is specified.

Figure 3-1. Valid CA Operand Specifications for Each 3704 and 3705 Channel Adapter Configuration.

```
                 {addr1}     {addr2}
          {(YES,[{NSC   }][, {NSC   }])}
[DYNADMP={       {NONE }     {NONE }  }  ]
          {NO                          }
```

Specifies whether the emulation program is to include the dynamic dump facility, which allows the storage contents of the communications controller to be transferred to the host processor without interrupting execution of the program, and specifies the subchannel(s) to be available for the transfer.

addr1 is the address of an emulation subchannel of the *first* type 1 or type 4 channel adapter (located in the base module of the communications controller) over which dynamic dump data can be transferred to the host processor.

addr2 is the address of an emulation subchannel of the *second* type 4 channel adapter (located in the first expansion module of a 3705) over which dynamic dump data can be transferred to the host processor.

NSC specifies that the native subchannel of the channel adapter is to be used for dump data transfer. (*NSC* in the first and second address positions refers to the native subchannel of the first and second channel adapters, respectively.)

NONE specifies that no subchannel of the channel adapter is to be used for dump data transfer. (*NONE* in the first and second address positions refers to the first and second channel adapters, respectively.)

Examples: (1) If the communications controller has a single type 1 or type 4 channel adapter and you wish to allow dynamic dump data to be transferred over an emulation subchannel, code DYNADMP= (YES, *addr1*). (2) If the controller has a type 4 channel adapter in the base module and a type 2 or type 3 adapter in the expansion module, code DYNADMP=(YES,NSC) to allow dump data transfer over the native subchannel of the type 4 channel adapter; code DYNADMP=(YES, *addr1*) to allow dump data transfer over an emulation subchannel of the type 4 adapter. (3) If the controller has two type 4 channel adapters, code DYNADMP=(YES, *addr1*, *addr2*) to allow dump data transfer over a specified emulation subchannel of each of the channel adapters; code DYNADMP=(YES,NONE, *addr2*) to allow transfer only over the specified subchannel of the second channel adapter; code DYNADMP=(YES,NONE,NSC) to allow transfer only over the native subchannel of the second adapter.

Rules governing which type of subchannel (native or emulation) can be used for transfer of dynamic dump data are as follows:

For type 1 channel adapter: The native subchannel (but not an emulation subchannel) can be used for dynamic dump data transfer.

For type 4 channel adapter: (1) An emulation subchannel can be used for dynamic dump data transfer. (2) The native subchannel can be used for dynamic dump data transfer if it is the subchannel over which the current emulation program was loaded.

Figure 3-2 shows, for each CA operand specification, the valid ways the DYNADMP operand can be used to specify subchannels to be used for transfer of dynamic dump data to the host processor.

If CA=	Then to allow dynamic dump data transfer over these subchannels:*		Code DYNADMP=
TYPE1	NSC_1		(YES,NSC)
TYPE4	NSC_1		(YES,NSC)
	ESC_1		(YES, *addr1*)
(TYPE4,TYPE4)	NSC_1		(YES,NSC)
	ESC_1		(YES, *addr1*)
	NSC_2		(YES,NONE,NSC)
	ESC_2		(YES,NONE, *addr2*)
	NSC_1 or NSC_2	‡	(YES,NSC,NSC)
	ESC_1 or ESC_2		(YES, *addr1*, *addr2*)
	NSC_1 or ESC_2	‡‡	(YES,NSC, *addr2*)
	ESC_1 or NSC_2	‡‡	(YES, *addr1*,NSC)

‡ Transfer of dynamic dump data is possible only over the channel adapter by which the emulation program was loaded.

‡‡ If the emulation program was loaded over the channel adapter represented by *addr1* or *addr2*, transfer of dynamic dump data is not possible over the channel adapter represented by NSC.

*ESC - emulation subchannel
NSC - native subchannel

Subscripts indicate channel adapter:
[1]first (base) adapter
[2]second (expansion) adapter

Figure 3-2. Valid Subchannel Address Specifications for Dynamic Dump Data Transfer

[JOBCARD={YES}] *(generation under OS only)*
{NO }

Specifies whether or not the program generation procedure is to provide a job card for the stage two input stream.

If you specify JOBCARD=YES or omit the operand, a job card is provided; if you specify JOBCARD=NO, you must provide your own job card.

This operand is optional.

[LESIZE=size] *(generation under OS only)*

Specifies the OS region size, in K (1024) bytes, to be used by all linkage editor job steps during stage two of program generation. *size* must exceed 200 and be less than 16,384 (16,384 K bytes).

If you omit the LESIZE operand, no REGION parameter appears on the EXEC cards for the linkage editor job steps and the system default region size is assumed. The system default region size is the amount of storage needed by the linkage editor to perform its task. The F-level linkage editor is required.

[LINETRC=[({YES}[,lines])]
{NO }

Specifies whether or not the channel and line interrupt line trace facility is to be included in the emulation program. The trace functions may be initiated from the control panel of the communications controller. Refer to the *Control Panel Guide* for more information on how to initiate the line trace from the control panel.

lines

Specifies the maximum number of lines that are to be traced (level 2) at any one time. This count is used to generate the BARSWAP table. For additional information on the BARSWAP table, refer to the *Program Reference Handbook.* The maximum number of lines that may be specified is 256. If *lines* is omitted, the count of the number of lines specified at emulation program generation is assumed.

{3705-2}
[MODEL={3705 }]
{3704 }

Specifies whether the generated program is to be loaded into and executed by a 3705-II (MODEL=3705-2), a 3705-I (MODEL=3705), or a 3704 (MODEL=3704). (A 3705-I may be specified as either MODEL=3705 or MODEL=3705-I.)

Note: This information is needed only by the generation procedure. The emulation program itself does not differ for the three machine types.

[NEWNAME={EP001 }
{symbol}

Specifies the name to be given to the generated emulation program.

Code NEWNAME=symbol, where *symbol* is any valid symbol that does not exceed eight characters. Alternatively, omit the operand or code NEWNAME=EP001.

```
[OPCSB2={YES}]
        {NO }
```

Specifies whether or not a 20-byte data buffer is to be provided for communication lines (1) that are serviced by a type 2 communication scanner, and (2) for which CHNPRI=HIGH is specified in the LINE macro. The 20-byte buffer, which is permanently assigned to the line, provides extra protection against overruns that can result from temporary slowdowns in channel operation or temporary peaks in data traffic in the network. Lines serviced by a type 2 scanner for which you do not specify OPCSB2=YES have two four-byte buffers.

If you omit this operand, OPCSB2=NO is assumed for lines associated with subchannels on a type 1 channel adapter, and OPCSB2=YES is assumed for the lines associated with subchannels on a type 4 channel adapter for which CHNPRI=HIGH is specified.

If you specify OPCSB2=YES, do not specify both CHNPRI=HIGH and TADDR=character in the same LINE macro. (Either of these operands, alone, may be specified.)

```
         {symbol}
[QUALIFY={NONE  }]
         {SYS1  }
```

(generation under OS only)

Specifies the first-level qualifier for OS data sets specified by the OBJLIB, LOADLIB, UT1, UT2, and UT3 operands of this macro. The data set name is formed by appending the characters *SYS1* or the characters you code in place of *symbol*, to the name specified by *dsname* in each of the preceding operands. The format is *xxx.dsname.*

`symbol`

Specifies the qualifier. This may be from one to eight alphanumeric characters; the first character must be alphabetic or $, #, or @. (Omit the period [.] that separates the qualifier and the data set name; the generation procedure appends the period to the qualifier you specify.)

`NONE`

Specifies that no qualifier is to be placed before the name specified by *dsname.*

`SYS1`

Specifies that SYS1 is to be used as the qualifier.

If this operand is omitted, SYS1 is assumed.

```
[TEST={YES}]
      {NO }
```

Specifies whether or not the emulation program is to include the panel-initiated line test function. The test routines may be initiated from the control panel of the communications controller only if you specify TEST=YES. (Refer to the *Control Panel Guide* for more information about the panel-initiated line test procedure.)

[TYPSYS={OS }]
        {DOS}

Specifies whether the emulation program is to be generated under OS or DOS.

[UNIT=unit type] *(generation under OS only)*

Specifies the type of device to be used for the assembler and linkage editor utility data sets during stage two of program generation under OS. You may specify either an actual device type (for example, UNIT=3330) or the name of a class of devices (for example, UNIT=SYSDA). The maximum number of characters you may specify is eight.

If you omit this operand, SYSSQ is assumed to be the unit type for the assembly steps and SYSDA is assumed for the linkage editing steps. The utility data sets for the linkage editor must reside on a direct-access device.

[UT1=dsname] *(generation under OS only)*

Specifies the name of a sequential OS data set to be used as work space for the assembly steps (SYSUT1). (The data set name may be simple or qualified, as determined by the QUALIFY operand of this macro. The simple data set name may be up to eight characters long; the first must be alphabetic or $, #, or @.) This data set must be preallocated and cataloged.

If you omit this operand, a temporary data set will be created during each assembly step using the type of device specified by the UNIT operand; the data set space provided is equivalent to SPACE=(1700,(800,800)).

[UT2=dsname] *(generation under OS only)*

Specifies the name of a sequential OS data set to be used as work space for the assembly steps (SYSUT2). (The data set name may be simple or qualified, as determined by the QUALIFY operand of this macro. The simple data set name may be up to eight characters long; the first must be alphabetic or $, #, or @.) This data set must be preallocated and cataloged.

If you omit this operand, a temporary data set will be created during each assembly step using the type of device specified by the UNIT operand; the data set space provided is equivalent to SPACE=(1700,(800,800)).

[UT3=dsname] *(generation under OS only)*

Specifies the name of a sequential OS data set to be used as work space for the assembly (SYSUT3) and linkage edit (SYSUT1) steps. (The data set name may be simple or qualified, as determined by the QUALIFY operand of this macro. The simple data set name may be up to eight characters long; the first must be alphabetic or $,#, or @.) This data set must be preallocated and cataloged.

If you omit this operand, SYSSQ is assumed to be the unit type for the assembly steps and the device type or the device class name specified in the UNIT operand is used for the linkage editing steps; the data set space provided is equivalent to SPACE=(1700,(800,800)).

Configuration Definition Macro Instruction

This section contains the single configuration definition macro instruction, CSB, required when defining an emulation program.

CSB Macro Instruction

Each communication scanner installed in the communications controller must be represented by a CSB macro. The CSB macro must follow the BUILD macro (or another CSB macro) in the emulation program source statements. It specifies:

- The type of communication scanner
- The internal oscillator (business-machine clock) rates for the scanner
- The location of the scanner within the controller
- The line address over which test data transmitted from a line interface being tested will be received.

Name	*Operation*	*Operands*
[symbol]	CSB	SPEED=(rate,...), WRAPLN=line addr [,MOD={n}] {0} {TYPE1} [,TYPE={TYPE2}] {TYPE3}

`[symbol]`

Provides a name for the macro. *symbol* may be any valid one to eight character assembler language symbol. The generation procedure does not check the symbol for validity.

`SPEED=(rate,...)`

Specifies the oscillator (business-machine clock) bit rates for up to four oscillators installed in the communication scanner. (Do not confuse this SPEED operand, which specifies the oscillator bit rates, with the SPEED operand of the LINE macro, which specifies the data rate for the communication line.) The speeds must be specified in the same order in which the oscillators are installed on the scanner, in ascending order according to speed. Standard oscillator bit rates are shown in Figure 3-3.

A type 3 scanner is always equipped with an oscillator that provides 150, 600, and 1200 bps bit rates, and may optionally have a 2000 or 2400 bps oscillator in addition. Therefore, if this CSB macro represents a type 3 scanner, specify SPEED=(150, 600, 1200) or SPEED=(150, 600, 1200, 2000) or SPEED=(150, 600, 1200, 2400), as appropriate.

Rate	*Represents*
45	45.5 bps
50	50.0
56	56.8 9
74	74.2
75	75.0
100	100.0
110	110.0
134	134.5
150	150.0
200	200.0
300	300.0
600	600.0
950	950.0
1200	1200.0
2000	2000.0
2400	2400.0

Figure 3-3. Standard Communication Scanner Oscillator Bit Rates

Note: If external (modem) clocking is used for any line attached to this communication scanner (CLOCKNG=EXT is specified in the LINE [or GROUP] macro), one of the oscillator bit rates you specify must be less than one-half of the lowest modem clocking rate specified in the SPEED operand of any LINE macro representing the attached lines.

This operand is required.

`WRAPLN=line addr`

Specifies the line interface address from which the controller will receive test data for a different line whose interface hardware is to be tested. (This function is the "wraparound test": test data supplied from the host processor or entered at the control panel of the controller is transmitted within the controller as far as the line set, then is looped back through the line set for a different line—the one whose address is specified in this operand—as a functional test of the line attachment hardware.)

The line specified need not be dedicated to the wraparound test operation; it can be any line that can be conveniently closed to normal teleprocessing operations when a wraparound test is needed. Both the specified line and the line to be tested must be closed to teleprocessing operations for the duration of the test. The online test (OLT) program selects the line to be tested.

line addr must be specified as the hexadecimal line address without the framing quotes (for example, 02F). This address must be within the range of line addresses on the communication scanner (see below) and must also be defined on a LINE macro. That is, the WRAPLN address *xxx* must have a corresponding LINE macro where ADDRESS=(*xxx,bb*). The subchannel address *bb* is then the subchannel address of the WRAPLN. If the appropriate LINE macro is not specified, assembly of the control blocks and tables results in an unresolved R-type constant. See Appendix B for an example of an emulation program using the WRAPLN operand.

Note: If any of the lines serviced by the scanner represented by this macro are BSC lines, the address you select for WRAPLN must be the line interface address for a BSC line.

This operand is required.

[MOD={n}]
{0}

Specifies the location of the communication scanner, as shown in Figure 3-4. The line interface addresses valid for each scanner type and module location are given. (The 3704 has only one module).

If scanner is in:	*Code MOD=*	*Line Interface Addresses for Type 1 Scanner*	*Type 2 Scanner*	*Type 3 Scanner*
3704	0	000-01F	020-03F	—
3705 base module	0	000-03F	020-05F	020-04F*
3705 first expansion module	1	—	0A0-0FF	0A0-0DF
3705 second expansion module	2	—	120-17F	120-15F
3705 third expansion module	3	—	1A0-1FF	1A0-1DF

*Type 3 scanner is not available in 3705-I base module; these addresses are valid only for a type 3 scanner in a 3705-II.

Figure 3-4. Location of Communication Scanners and Valid Line Interface Addresses

{TYPE1}
[TYPE={TYPE2}]
{TYPE3}

Specifies whether the communication scanner is type 1, type 2, or type 3.

Valid designations for scanner type are:

If controller is 3704 (MODEL=3704):

TYPE1
TYPE2

If controller is a 3705-I (MODEL=3705):

TYPE1 (valid only for MOD=0)
TYPE2
TYPE3 (valid only for MOD=1, MOD=2, or MOD=3)

If controller is a 3705-II:

TYPE2
TYPE3

If you omit this operand and you have specified MODEL=3704 in the BUILD macro, the scanner is assumed to be type 1. If you specified MODEL=3705 or MODEL=3705-2 (or omitted the MODEL operand), the scanner is assumed to be type 2.

Teleprocessing Network Configuration Macro Instructions

This section contains the teleprocessing network configuration macro instructions—GROUP and LINE—to be used in specifying the communication lines attached to be communications controller.

The GROUP macro represents a grouping of lines having certain characterictics in common. The grouping is referred to as a *physical* line group because it consists of lines having certain "physical" attributes in common, such as the type of stations attached to them.

The LINE macro represents one start-stop or binary synchronous (BSC) communication line attached to the communications controller. All LINE macros representing lines in a physical line group must appear between the GROUP macro representing that group and the next GROUP macro. Assume, for example, that the first two lines in a configuration have similar attributes allowing them to be in the same line group but that the third line has different characteristics which require it to be in a different line group. (A single line can constitute a line group). The macro sequence in this case would be as follows:

```
GROUP
  LINE
  LINE
GROUP
  LINE
```

If all the lines were dissimilar, each of the three LINE macros would have to be preceded by a GROUP macro. For example, a configuration having one start-stop TWX line, one start-stop 2260 line, and one BSC line would require three line groups, each consisting of one line. On the other hand, a configuration comprising three similar lines (for example, three BSC lines) could (but not necessarily) be in the same line group, with only one GROUP macro immediately preceding the first LINE macro.

Most of the characteristics that pertain to the lines in a group can be specified in the GROUP macro instruction. This makes it desirable to include similar lines in a single group. It is not necessary that a characteristic be identical for all of the lines in a group in order to specify that characteristic in the GROUP macro. You may code the exceptions on an individual LINE macro. Any characteristic that you code in the LINE macro overrides the characteristic specified in the GROUP macro. If, for example, all except one of the six lines in a group have the same characteristic in common, you could specify the common characteristic in the GROUP macro and the differing characteristic in the LINE macro for that exceptional line. If, however, you do not define the characteristic in either macro, the default value specified in the LINE macro description is assumed.

A summary of the operands for the GROUP and the LINE macro instructions is given in Figure 3-5. The figure indicates the operands that can be specified for each macro instruction and that apply to either start-stop or BSC lines. You may not specify the ADDRESS, DUALCOM, and AUTO operands in the GROUP macro or the CHAREC, DELAY, DIAL, EOB, EOT, LNCTL, REPLYTO and TEXTTO operands in the LINE macro.

Note: The start-stop and 2260 or 2265 terminals in your network require a GROUP macro with LNCTL=SS. The BSC terminals in your network require a GROUP macro with LNCTL=BSC. Networks with both start-stop and BSC line controls coded in this fashion permit you to have a more efficient grouping of similar characteristics.

Operand:	*GROUP Macro*	*LINE Macro*	*Start-Stop*	*Binary Synchronous*
ADDRESS		•	•	•
AUTO		•	•	•
BUFSIZE	•	•		•
CHAREC	•		•	
CHECK	•	•	•	
CHNPRI	•	•		•
CLOCKNG	•	•	•	•
CODE	•	•		•
CU	•	•	•	•
CUTYPE	•	•	•	•
DATRATE	•	•	•	•
DELAY	•		•	
DIAL	•		•	•
DISABLE	•	•		•
DUALCOM		•		•
DUPLEX	•	•	•	•
EOB	•		•	
EOT	•		•	
FEATURE*	•	•		
INTPRI	•	•	•	•
LNCTL	•		•	•
MODEM	•	•	•	•
MULTI	•	•	•	
NEWSYNC	•	•		•
PAD	•	•		•
QUIET	•	•	•	
REPLYTO	•		•	
RING	•	•	•	•
SPEED	•	•	•	•
TADDR	•	•		•
TERM	•	•	•	•
TEXTTO	•		•	
UNITXC	•	•	•	

*The FEATURE operand has both start-stop and BSC suboperands.

Figure 3-5. Summary of Operands for the GROUP and LINE Macro Instructions

GROUP Macro Instruction

The GROUP macro instruction represents a grouping of lines having certain characteristics in common. The grouping is referred to as a *physical line group* because it contains lines having certain "physical" attributes in common, such as the following:

- All lines in the group are either nonswitched or switched (lines in a non-switched group may be point-to-point or multipoint or a combination of the two).
- All stations connected to lines in the group are start-stop, or all are binary synchronous.

No line may be included in more than one line group.

For each line group, one GROUP macro is required.

The GROUP macro indicates the beginning of a sequence of LINE macros for lines within the group, and specifies:

- Whether the lines are switched or nonswitched.
- Whether the lines are used for start-stop (asynchronous) or binary synchronous communication.
- Optional or variable characteristics that all lines in the group must have in common.
- Certain procedural options to be applied to all lines in the group.

Name	*Operation*	*Operands*
symbol	GROUP	{[XONOFF][,chars]} [,CHAREC=({XON,chars })] {XOFF,chars } {NO,chars } [,DELAY={600 }] {1200} {NO } [,DIAL={YES}] {NO } [,EOB=(character[,F])] [,EOT=(character[,F])] [,LNCTL={SS }] {BSC} [,QUIETCT= { count }] {0 } [,REPLYTO={count}] {3.0 } [,TEXTTO={count}] {25.6 }

`symbol`

Provides a name for the line group and is required. *symbol* may be any valid assembler-language symbol; the first character may not be $.

A line group name is required.

```
        {[XONOFF][,chars]}
[,CHAREC={XON,chars      })]
         {XOFF,chars     }
         {NO,chars       }
```

(teletypewriter lines only)

Specifies the message ending sequence to be recognized by the emulation program when receiving from an AT & T 83B3, WU 115A, or WU TWX teletypewriter terminal.

(Upon recognizing the specified ending sequence received from a line, the program signals Channel End, Device End and Status to the emulation subchannel associated with the line.)

Specify this operand as follows:

For TWX terminals:

If you wish the program to recognize, as an ending character or sequence:

- Either XON (transmit-on) or XOFF (transmit-off), but no other character, specify CHAREC=XONOFF or omit the operand. (On some TWX terminals XON is called DC1 (device control 1) and XOFF is called DC2.)
- Either XON followed by one or two characters or XOFF followed by one or two characters, specify CHAREC=(XONOFF,chars). Specify *chars* as the hexadecimal representation of the characters that the terminal sends.
- Only XON or one or two user-selected characters, specify CHAREC=(XON,chars).
- Only XOFF or one or two user-selected characters, specify CHAREC=(XOFF,chars).
- Only one or two characters that you designate, but not XON or XOFF, specify CHAREC=(NO,chars).

Coding CHAREC=(,chars) is equivalent to coding CHAREC=(XONOFF,chars).

Note: The program will not recognize WRU (who-are-you) or ENQ (enquiry) characters (A1), or any other characters other than XON or XOFF, unless you specify the required characters in this operand. WRU unconditionally ends a Read command with Channel End and Device End.

For 83B3 and 115A terminals:

These two types of terminals do not receive or transmit XON and XOFF characters. The standard ending sequence for such terminals is FIGS H LTRS. The program will recognize this sequence if you specify CHAREC=XONOFF or omit the operand. [CHAREC=(XONOFF,chars) is *not* valid.]

If the terminals in your network send an ending character or sequence other than the standard FIGS H LTRS sequence, specify the character or sequence sent as CHAREC=(,chars). *chars* must be the hexadecimal representation of the transmission code bit patterns of the required characters.

```
       {600 }
[DELAY={1200}]
       {NO  }
```

(World Trade teletypewriters only)

Specifies whether the World Trade teletypewriter (teleprinter) terminals attached to lines in the group represented by this GROUP macro require a line turnaround time of 70 to 80 milliseconds.

Specify DELAY=600 if the terminals require the delayed turnaround and operate at 600 bps; specify DELAY=1200 if the terminals require the delayed turnaround and operate at 1200 bps.

If delayed turnaround is not required, specify DELAY=NO or omit the operand.

[DIAL={YES}]
      {NO }

Specifies whether or not the lines in the group are switched. If they are switched, code DIAL=YES. If they are not, code DIAL=NO or omit the operand.

Note: DIAL=NO must be specified if a manual *Data Access Arrangement* (DAA) is used and the call is manually originated and answered.

[EOB=(char[,F]

(World Trade teletypewriters only)

Specifies the character sequence the program is to recognize at the end-of-block (end-of-message) sequence when received from any World Trade teletypewriter (teleprinter) or certain (specially modified) U.S. and Canadian teletypewriter terminals. These terminals typically send, as an end-of-message indication, either a sequence of four identical characters or FIGS character LTRS.

If the terminal transmits a four-character sequence, specify the character used as EOB=char, where *char* is the hexadecimal representation of the character transmitted. *Example:* If the terminal sends the sequence MMMM (in letters shift), and the hexadecimal representation of the letter M in transmission code is 1C, you would code EOB=1C.

If the terminal transmits the sequence FIGS character LTRS, code *char* as the hexadecimal representation of the character sent and also code the F following the character. *Example:* If the terminal sends the sequence FIGS M LTRS and the hexadecimal representation of the letter M is 3C (in figures shift), you would code EOB=(3C,F).

Note: Appendix G lists the transmission code bit pattern for the ITA2 and ZSC3 codes.

The end-of-block (end-of-message) sequence may be specified in either the EOB or the CHAREC operand, but not both.

[EOT=(char[,F])]

(World Trade teletypewriters only)

Specifies the character sequence the program is to recognize as the end-of-transmission sequence when received from any World Trade teletypewriter terminal or certain (specially modified) U.S. and Canadian teletypewriter terminals. These terminals typically send, as an EOT indication, either a sequence of four identical characters or FIGS character LTRS.

If the terminal transmits a four-character sequence, specify the character used as EOT=char, where *char* is the hexadecimal representation of the character transmitted. *Example:* If the terminal sends the sequence AAAA (in letters shift), and the hexadecimal representation of the letter A in transmission code is 18, you would code EOT=18.

If the terminal transmits the sequence FIGS character LTRS, code *char* as the hexadecimal representation of the character sent and also code the F following the character. *Example:* If the terminal sends the sequence FIGS A LTRS as the end-of-transmission sequence, and the hexadecimal representation of the letter A is 38 (in figures shift), you would code EOT=(38,F).

The standard teletypewriter ending sequence is FIGS H LTRS. If you omit the CHAREC and EOT operands, this is the sequence the program recognizes as the EOT sequence when receiving from a teletypewriter terminal.

Note: Appendix G lists the transmission code bit pattern for the ITA2 and ZSC3 codes.

[LNCTL={SS }]
{BSC}

Specifies whether the line group consists of start-stop lines (LNCTL=SS) or binary synchronous lines (LNCTL=BSC).

[QUIETCT= {count}]
{0 }

(start-stop lines only)

Specifies the number of character times that the emulation program will allow to elapse between the end of a receive operation and the beginning of a transmit operation on a start-stop line. The elapsed time allows the line to electrically "quiesce" following the the receive operation. (The line must become electrically quiet before the next data transmission begins or loss of message data may result.)

Note: The interval (number of character times) following a normal receive operation equals the value you specify (from 0 to 10) plus 2. The interval following receipt of a negative response to polling equals the value you specify. Thus, if you specify QUIETCT=5, a normal receive operation is followed by seven character times and a negative response to polling is followed by five character times before the next transmission begins.

The default value of 0 is appropriate for most start-stop lines operating at speeds under 1200 bps. For lines operating at 1200 bps or more, one or more extra character times may be necessary to ensure quieting of the line. The recommended value for 1200 bps start-stop lines is five (QUIETCT=5).

The minimum you may specify is 0; the maximum is 10.

[REPLYTO={count}]
{3.0 }

(start-stop lines only)

Specifies the reply time-out value, in seconds, for the lines in the line group. If at the expiration of this interval the emulation program has not received from the station a response to polling or selection, or to message text, it makes no further attempt to communicate with the station. Instead, it indicates that a time-out error has occurred.

You may specify this value as an integral number of seconds or as seconds and tenths of seconds.

If you specify REPLYTO=0.0, no time-out occurs regardless of the time that elapses between sending to the station and receiving the response.

The maximum value is 51.1 seconds.

This operand is valid only for start-stop lines (LNCTL=SS is specified in [or LNCTL operand omitted from] this GROUP macro).

If you omit the operand, the program observes a value of 3.0 seconds.

Note: Diagnostic programs for IBM 2845 and 2848 control units attached to lines within the group wili fail if you specify a value of more than 2.0 seconds in the REPLYTO operand, or if you omit the operand.

`[TEXTTO={count}]`
`        {25.6 }`

(start-stop lines only)

Specifies the text time-out value, in seconds, for the lines in the line group. If the interval between any two successive message characters received from a station exceeds this value, the program ends the operation with a text time-out error indication.

You may specify this value as an integral number of seconds or as seconds and tenths of a second.

If you code TEXTTO=0.0, no time-out occurs regardless of the time interval that elapses between receipt of successive characters.

The maximum value is 51.1 seconds.

If you omit this operand, the program observes a timeout of 25.6 seconds.

This operand is valid only for start-stop lines (LNCTL=SS is specified in [or LNCTL operand omitted from] this GROUP macro).

Note: Diagnostic programs for IBM 2845 and 2848 control units attached to lines within the group will fail if you specify a value of more than 2.0 seconds in the TEXTTO operand, or if you omit the operand.

LINE Macro Instruction

The LINE macro represents one start-stop or BSC communication line attached to the communications controller and specifies:

- Whether the communications facility of which the line is a part is a half-duplex or a duplex facility.
- The address, within the communications controller to which the line is attached, and the corresponding CPU subchannel address(es) over which the access method communicates with stations attached to the line.
- The priority of the emulation subchannel associated with this line, with respect to other emulation subchannels.
- The address, within the communications controller, to which the related automatic calling unit (ACU), if any, is attached.
- The speed of the line—that is, the rate (in bits per second) at which the controller and stations on the line transmit data.
- The interrupt priority of the line.
- Whether the modem or the communication scanner is to provide clocking of the communication line.
- Whether the modem by which the line is attached to the controller, if a dual data rate modem, is to operate at the higher or lower of the two rates.
- Whether (for a switched line) the modem operates in ring indicator mode.
- The type of transmission control unit (IBM 2701, 2702, 2703) that the program is to emulate when communicating with stations over the line represented by this LINE macro.
- The dual communications interface address, if any, associated with the line (applicable only if the transmission control unit [2701] being emulated is equipped with the dual communications interface feature).
- Certain features of the transmission control unit being emulated.
- The type of start-stop or BSC station(s) connected to the line.
- Certain features with which the stations attached to the line are equipped.
- The transmission code used by stations on the line (if BSC stations).

Name	*Operation*	*Operands*
symbol	LINE	ADDRESS=(line addr,subchan addr1[,subchan addr2,...subchan addrn]), SPEED=rate [,AUTO={address}] {NONE } [,BUFSIZE=n] [,CHECK={DCD }] {NODCD} [,CHNPRI={NORMAL}] {HIGH } [,CLOCKNG={EXT}] {INT} [,CODE=(code1[,code2])] {2701} [,CU={2702}] {2703}

Name	*Operation*	*Operands*
		{2845} {2848} [,CUTYPE={2972}] {3271} {3275} [,DATRATE={HIGH}] {LOW } [,DISABLE={YES}] {NO } [,DUALCOM={(line addr,{A})}] {B} NONE [,DUPLEX={HALF}] {FULL} [,FEATURE=([{DUALCODE}] [,{IMEND }] {NODUALCD} {NOIMEND} [,{LRC }] [,{SPACE }])] {NOLRC} {NOSPACE} {0} [,INTPRI={1}] {2} {3} [,MODEM={OPTION1}] {OPTION2} {NTT } [,MULTI={YES}] {NO } [,NEWSYNC={YES}] {NO } [,PAD={YES}] {NO } [,QUIET={YES}] {NO } [,RING={YES}] {NO } [,TADDR={character}] {NONE } [,TERM=type] [,UNITXC={YES}] {NO }

`symbol`

Provides a name for the communication line and is required. *symbol* may be any valid assembler language symbol; the first character may not be $.

`ADDRESS=(line addr,subchan addr1[,subchan addr2,...,subchan addrn])`

Specifies the line interface address and the corresponding emulation subchannel address(es) for the line represented by this LINE macro.

Specify *three* hexadecimal digits for the line interface address. Specify *two* hexadecimal digits, each followed by -0 or -1, for each subchannel address: -0 to indicate that the associated channel adapter is installed in the *base* module, -1 to indicate that the associated adapter is installed in the first *expansion* module (3705 only). (-0 may be omitted, in which case the base module is assumed as the adapter location.) *Examples:* (1) ADDRESS=02F,2E-0: line address 02F associated with subchannel address 2E via channel adapter in base module; (2) ADDRESS=(02F,2E-0,17-1,1C-0): line address 02F associated with subchannel addresses 1C and 2E via channel adapter in base module and with subchannel address 17 via adapter in expansion module.

The subchannel address must be associated with a type 1 or type 4 channel adapter. The subchannel address specified for this line must not be the same as a subchannel address associated with any other line or used by any other program facility (such as the dynamic dump utility).

Notes: (1) More than one subchannel address is valid only if the controller in which the program is to be executed has one or two type 4 channel adapters; (2) if the specified line address is for a line serviced by a type 3 communication scanner, the subchannel addresses must be associated with a type 4 channel adapter.

The subchannel address(es) must be within the range specified by the HICHAN and LOCHAN operands of the BUILD macro for the associated channel adapters (base and expansion modules).

Figure 3-6 gives the range of valid line addresses for this operand.

This operand is required.

Note: *subchan addr* is the subchannel address at which the access method logically addresses the line. The same subchannel address can be specified for two different lines only if the dual communications interface option is specified for the pair of lines (via the DUALCOM operand of the LINE macro).

If line is attached to:	*Range of valid addresses is:*
Type 1 scanner in 3704	000-01F
Type 2 scanner in 3704	020-03F
Type 1 scanner in 3705	000-03F
Type 2 scanner in 3705	020-05F 0A0-0FF 120-17F 1A0-1FF
Type 3 scanner in 3705	020-04F 0A0-0DF 120-15F 1A0-1DF

Figure 3-6. Valid Line and Auto Call Interface Addresses for LINE Macro Instruction

SPEED=rate

Specifies the data rate for this line in bits per second; that is, the rate at which the stations on the line transmit to the communications controller, and vice versa.

If internal (business-machine) clocking is used (see the CLOCKNG operand), the rate must be one of the four oscillator rates specified for the communication scanner to which this line is attached (SPEED operand of the CSB macro). Specify the line speed in bits per second, omitting a fractional part, if any. For example, specify a line speed of 1200 bps as SPEED=1200; specify a line speed of 134.5 bps as SPEED=134 (omitting the decimal point and fraction).

If external (modem) clocking is used, the rate must be the clocking rate of the modem by which the line is attached to the communications controller (which is not necessarily one of the oscillator bit rates specifed for the scanner). However, the SPEED operand of the CSB macro must specify a scanner oscillator bit rate *less than one-half* of the modem clocking rate you specify in this SPEED operand.

The maximum speed you may specify if the line is connected by a type 1 communication scanner is 7200 bps; the maximum if the line is connected to a type 2 or type 3 scanner is 56000 bps.

This operand is required.

[AUTO={line addr}]
{NONE}

Specifies whether the auto call facility is present for this line and gives the automatic calling unit (ACU) interface address. This may be determined from the system designer.

If the line is so equipped, code the ACU interface address in the operand (in hexadecimal, without framing ["'] characters). Example: AUTO=020.

If the line does not have the ACU facility, code AUTO=NONE or omit the operand.

Figure 3-6 gives the range of valid addresses.

If this operand is specified, DIAL=YES must be specified in the GROUP macro.

[BUFSIZE=n] *(lines serviced by type 3 scanner only)*

Specifies (1) the size of the emulation mode buffers for a line serviced by a type 3 communication scanner (each such line has two buffers of the specified size for data being sent to or received from the line), and (2) the amount of data (up to 32 bytes) transferred over the channel to the host processor without program interrupt.

The value of *n* may be any of the following:

4 (see note below)	96
8	128
16	160
32	192
64	224

For a given amount of data passing over the line, use of larger buffers affords more protection against possible overruns than do smaller buffers. (Overruns can result from temporary slowdowns of channel operation or from momentary peaks

in data traffic through the network.) Use of larger buffers also results in less interrupt-processing overhead for line operations and—up to 32 bytes—less interrupt-processing overhead for channel operations. The amount of data transferred across the channel is equal to *n* up to 32 bytes. For values of *n* exceeding 32, the amount of data transferred over the channel is 32 bytes.

If you omit this operand, two 32-byte buffers (*n* =32) are provided if the line speed you specify is 19,200 bps or less; two 64-byte buffers (*n* =64) are provided if the specified line speed exceeds 19,200 bps.

Note: Do not specify 4-byte buffers (*n* =4) if the line represented by this LINE macro is polled (POLLED=YES) and the size of the poll entries in the service order table is six bytes or more. (A poll entry comprises the address characters, ENQ, and index byte used to poll one station on the line [for example, Cl Cl 40 40 ENQ Index).

This operand is valid only for a communication line serviced by a type 3 scanner.

`[CHECK={DCD   }]`
`      {NODCD}`

(switched, duplex, start-stop lines only)

Specifies whether the controller is to use the 'data carrier detect' option for the line represented by this LINE macro. Use of this option prevents access to an application program's data by a station that dials the controller over this line at the moment the existing connection to a station is lost. Continuous monitoring of the 'data carrier detect' signal from the modem gives positive assurance that the switched line connection is still established.

CHECK=DCD is valid only if the LINE macro specifies DUPLEX=FULL and the GROUP macro specifies LNCTL=SS and DIAL=YES.

`[CHNPRI={NORMAL}]`
`        {HIGH  }`

(BSC lines only)

Specifies the priority of the subchannel associated with this line with respect to the other subchannels.

This operand is valid only for a BSC line (LNCTL=BSC is specified in the GROUP macro).

CHNPRI=NORMAL is ordinarily appropriate unless the line represented by this LINE macro is to operate at a data rate of 19,200 bps or more, and the majority of the remaining lines are slow speed lines (2,400 bps or less).

Note: If OPCSB2=YES is specified in the BUILD macro, or in that macro the CA operand specifies a type 4 channel adapter, then all BSC lines serviced by a type 2 scanner and specified as requiring high channel priority (CHNPRI=HIGH) are provided with additional buffering. (See the description of the OPCSB2 operand.)

Do not specify CHNPRI=HIGH if you specify OPCSB2=YES in the BUILD macro and TADDR=char in this LINE macro.

See the *EP Storage and Performance* manual for additional information about subchannel priorities.

`[CLOCKNG={EXT}]`
`         {INT}`

Specifies whether the modem (data set) or the communication scanner is to provide clocking for the line. This may be determined from the system designer.

CLOCKNG=INT specifies that the scanner provides clocking.
CLOCKNG=EXT specifies that the modem (whether external to or contained within the controller) provides clocking.

If this LINE macro represents a BSC line (LNCTL=BSC), CLOCKNG=EXT is assumed if you omit this operand.

If this LINE macro represents a start-stop line (LNCTL=SS), CLOCKNG=INT is assumed if you omit this operand.

Note: Notice that the letter I is omitted from the CLOCKNG operand.

```
[CODE=(code1[,code2])]
```

Specifies the transmission code with which the access method will communicate with stations over the BSC line represented by this LINE macro.

code1 and *code2* each may be specified as EBCDIC, USASCII, or USASCII-T. If you specify USASCII, LRC checking is performed for all data transmitted and received over the line. If you specify USASCII-T (transparent USASCII), CRC checking is performed for all data (transparent or nontransparent) transmitted and received over the line.

Specify *code2* following *code1* if the dual code interface facility is present for the line (that is, FEATURE=DUALCODE is specified for the line).

If you specify FEATURE=DUALCODE and omit the *code2* parameter, *code2* is assumed to be (1) USASCII, if you specify *code1* as EBCDIC or omit the CODE operand; (2) EBCDIC, if you specify *code1* as USASCII or USASCII-T.

Note: USASCII-T is valid only if the line represented by this LINE macro is serviced by a type 3 communication scanner.

```
    {2701}
[CU={2702}]
    {2703}
```

Specifies whether the transmission control unit functions that the emulation program is emulating for this line are those of an IBM 2701, 2702, or 2703.

Code the value corresponding to the type of TCU to which the line represented by this LINE macro was formerly attached (that is, prior to installation of the communications controller in place of the TCU).

Note (for DOS users only): If the program is emulating a 2701 and the access method is under DOS, code CU=2702 or 2703 to avoid Test I/O problems. The subchannel with which this line is associated must be defined to the DOS system as the control unit specified in the CU operand.

```
        {2845}
        {2848}
[CUTYPE={2972}]
        {3271}
        {3275}
```

Specifies whether the control unit of the cluster type station(s) attached to this line is an IBM 2845, 2848, 2972, 3271, or 3275.

CUTYPE=2845 or CUTYPE=2848 is valid only if LNCTL=SS is specified in the GROUP macro for this line. CUTYPE=2972 or CUTYPE=3271 or CUTYPE=3275 is valid only if LNCTL=BSC is specified.

Note: If more than one control unit type is attached to this line, specify either type in the CUTYPE operand. For example, if both 3271s and 3275s are attached to the line, specify either CUTYPE=3271 or CUTYPE=3275.

```
[DATRATE={HIGH}]
         {LOW }
```

Specifies at which of two data rates a dual-rate modem is to transmit. (Determine this from the system designer.)

Code DATRATE=HIGH if the higher rate is to be used. Code DATRATE=LOW (or omit the operand) if the lower rate is to be used.

Note: DATRATE=HIGH is invalid for modems attached to lines sets 1A, 1B, 1C, 2A, 3A, 4A, 4B, and 4C, and if specified may cause a feedback-check error condition.

If the modem by which this line is attached to the communications controller has only one data rate, specify DATRATE=LOW or omit the operand.

```
[DISABLE={YES}]
         {NO }
```

Specifies whether or not the World Trade modem for the line represented by this LINE macro requires a "long disable time-out" when disconnecting from the line. The time-out provided is 25.6 seconds if you specify DISABLE=YES.

For World Trade modems that do not require the long time-out, DISABLE=NO is the appropriate value. If in doubt, consult the supplier or installer of the modem.

```
[DUALCOM={(line addr,{A})}]
         {      {B}}
         {NONE       }
```

Specifies that the emulation program is to emulate the dual communications interface function for the line represented by this LINE macro. Specify this operand only if the transmission control unit (IBM 2701) to which the line was formerly attached (that is, before replacement of the 2701 by the communications controller) was equipped with the dual communications interface feature.

`line addr`

Specifies the controller line interface address to which the *alternate* line (of the two lines formerly attached to the 2701 dual communications interface) is attached.

`A`

Specifies that *this* line (whose address is specified in the ADDRESS operand of this LINE macro) corresponds to dual communications interface *A*.

`B`

Specifies that *this* line (whose address is specified in the ADDRESS operand of this LINE macro) corresponds to dual communications interface *B*.

NONE

Specifies that the 2701 being emulated is not equipped with the dual communications interface for this line.

Example: Assume that two communication lines formerly attached to an IBM 2701 line address via the dual communications interface are now attached to the communications controller line addresses 020 and 021, and that these two lines when attached to the 2701 corresponded to dual communications interface *A* and *B*, respectively. Assume that the access method in the host processor is to communicate with these two lines, alternately, via emulation subchannel 017.

You would specify the respective LINE macros for the two lines such that (1) the ADDRESS operands of both specify the same emulation subchannel, 017; (2) each DUALCOM operand specifies the line address designated in the ADDRESS operand of the *other* LINE macro; and (3) the DUALCOM operand of each macro specifies the interface—*A* or *B*—to which the lines formerly corresponded, when attached to the 2701:

```
L1   LINE    ADDRESS=(020,17),DUALCOM=(021,A)
L2   LINE    ADDRESS=(021,17),DUALCOM=(020,B)
```

[DUPLEX={HALF}]
{FULL}

Specifies whether the communication line and modems constitute a half-duplex or (full-) duplex facility. Determine from the system designer the appropriate value to code.

If this operand is omitted and CUTYPE=2845 or 2848, or TERM=2260 or 2265 has been specified, the default for this operand is DUPLEX=FULL; otherwise, the default is DUPLEX=HALF.

Note: This should not be confused with half-duplex data transfer. This operand specifies only the physical characteristic of the communication facility (line and modems). All data transfer between the stations supported by the emulation program occurs only in half-duplex mode, regardless of whether the line is half-duplex or duplex.

[FEATURE=(...)]

Specifies the machine features with which certain types of stations may be equipped.

[{DUALCODE}]
{NODUALCD}

(BSC lines only)

Specifies whether or not the program is to emulate, for the line represented by this LINE macro, the dual code feature of an IBM 2701. (The dual code feature allows message transmission over the line in either of two transmission codes, EBCDIC or USASCII (the latter specified in the CODE operand as USASCII or USASCII-T), as selected by command from the access method.) Specify DUALCODE if you wish to allow either code to be used on the line represented by this LINE macro, and if the access method is capable of changing the code. Specify NODUALCD if the dual code function is not required.

If you omit the FEATURE operand or DUALCODE, DUALCODE is assumed if *code2* is specified in the CODE operand. (See the explanation of the CODE operand.)

The transmission code specified in the CODE operand for the line represented by this LINE macro is assumed to be the primary code.

[{IMEND }]
 {NOIMEND}

(specially-equipped start-stop terminals only)

Specifies whether or not the program is to delay ending a receive operation for a line upon recognizing an EOT character or sequence sent by a start-stop terminal. If you specify NOIMEND in the FEATURE operand, or omit the parameter, the program delays ending the receive operation until the line becomes electrically 'quiet' following receipt of the EOT. The absence of further characters indicates that the EOT is valid and not a data character converted by line noise to a spurious EOT. This is appropriate for most applications. However, if your application requires immediately ending the receive operation upon detecting the EOT, specify IMEND in the FEATURE operand.

[{LRC }]
 {NOLRC}

(start-stop terminals only)

Specifies whether or not the start-stop terminals connected to the line represented by this LINE macro are equipped with record checking capability (either as an inherent function or as a feature).

LRC is the appropriate value if the station with which the controller communicates over this line is one of the following:

IBM 1050	1050
IBM 1060	1060
IBM 2740 Model 1	2740-1
IBM 2740 Model 2	2740-2
IBM 2260	2260
IBM 2265	2265
IBM System/7	SYS7

[{SPACE }]
 {NOSPACE}

(teletypewriter terminals only)

Specifies whether or not the emulation program is to react to space characters received from 83B3, 115A, or World Trade teletypewriter (teleprinter) terminals as downshift characters. If you specify SPACE, each space character received from a terminal causes the program to send all subsequent text characters to the host processor in their downshifted form.

If you specify NOSPACE (or omit the parameter), the program does not convert the characters to their downshifted form, but instead sends them as received from the terminal.

[INTPRI={0}]
        {1}
        {2}
        {3}

Specifies the interrupt priority for this line relative to other lines attached to the same communication scanner. Priority 3 is the highest and 0 is the lowest. Lines with high data rates and lines serviced by a type 3 communication scanner should be assigned higher priorities than lines with lower data rates and lines serviced by a type 1 or type 2 scanner.

If this line is serviced by a type 2 or type 3 scanner, the valid range for INTPRI is 0 through 3. If this line is attached to a type 1 communication scanner, the only valid values are 0 and 1, with 1 being the higher priority. Appendix H gives a method for determining the interrupt priorities for each line in the network.

```
       {OPTION1}
[MODEM={OPTION2}]
       {NTT    }
```

Specifies whether the communication line (1) is enabled immediately after the communications controller has been loaded (IPL) or the system reset key has been pressed (MODEM=OPTION1); or (2) is disabled after the controller has been loaded or the System Reset key has been pressed, and must subsequently be enabled by command from the access method (MODEM=OPTION2). (Until the line is enabled, most commands issued to the line by the access method will result in an Intervention Required indication to the access method.)

Which of these two procedures is appropriate depends upon whether or not the 'data set ready' lead within the modem that attaches the line to the controller is continuously activated. This may be learned from the supplier or installer of the modem.

If the 'data set ready' lead of the modem is continuously activated, specify MODEM=OPTION1. This choice is valid for lines to which IBM 1030, 1050, 1060, 2740 Model 1 or 2, 2741, or System/7 stations are attached.

If the 'data set ready' lead is *not* continuously activated (and the access method consequently must issue an enable command to enable the line), specify MODEM=OPTION2 or omit the operand.

Note: Enable and Disable commands are not applicable to lines to which IBM 2845 or 2848 control units are attached. Therefore such lines are always enabled when the controller is loaded or the system reset key is pressed; it is necessary to specify MODEM=OPTION1.

In Japan, code MODEM=NTT if the modem that attaches the line to the controller is supplied by Nippon Telephone and Telegraph.

```
[MULTI={YES}]
       {NO }
```

(IBM 2845, 2848 only)

Specifies whether the line represented by this LINE macro connects the controller to (1) one or more 2845 or 2848 display controls equipped with multipoint line control (MULTI=YES); or (2) a single display control equipped with point-to-point line control (MULTI=NO).

```
[NEWSYNC={YES}]
         {NO }
```

(BSC lines only)

Specifies whether or not the communications controller is to supply the 'new sync' signal to the modem (data set) used by this line.

```
[NEWSYNC={YES}
         {NO }
```

(BSC lines only)

Specifies whether or not the communications controller is to supply the "new sync" signal to the modem (data set) used by this line.

NEWSYNC=YES is valid only if the modem at the controller serving the line has the new sync feature *and* if the communications controller is the multipoint master (not tributary) station for a duplex (*not* half-duplex) line on which multipoint line control is used.

If you omit the NEWSYNC operand, NEWSYNC=YES is assumed if you specify LNCTL=BSC, DUPLEX=FULL, CLOCKNG=EXT, and DIAL=NO. (All operands must be so specified.) If you omit the operand and specify LNCTL=BSC, but any of the three remaining operands is now as shown, NEWSYNC=NO is assumed. For LNCTL=SS, the NEWSYNC operand has no meaning.

Determine from your IBM representative or the installer or supplier of the modem (if other than an IBM modem) whether the appropriate conditions above prevail. If they do not, the newsync function cannot be used, and you must therefore specify NEWSYNC=NO.

`[PAD={YES}]`
`     {NO }`

(binary synchronous lines only)

Specifies whether or not the communications controller, when emulating an IBM 2703 Transmission Control unit, is to verify that the first four bits of trailing pad characters received from the line are all 1's, that is, hexadecimal 'F'. If you specify PAD=YES or omit the operand, the controller checks each pad character received and indicates a data check error if the first four bits are not all 1's.

If you specify PAD=NO, the controller, when emulating a 2703, does not check pad characters in this manner.

`[QUIET={YES}]`
`       {NO }`

Specifies whether or not the program is to observe a "long line quiet" time-out of 25.6 seconds when receiving from the line represented by this LINE macro. If you specify QUIET=YES, the program observes the long time-out. If you specify QUIET=NO (or omit the operand), the normal time-out of 3.0 seconds is observed.

QUIET=YES should not be specified if FEATURE=IMEND is specified for the line represented by this LINE macro.

`[RING={YES}]`
`      {NO }`

(not applicable to U.S. and Canada)

Specifies whether or not the ring indicator mode of the automatic answer operation is to be used for this line. This depends solely upon the type of modem that connects the line to the controller. Determine from the modem supplier or installer whether it has a 'ring indicator interface' lead.

If it has the ring indicator interface lead, code RING=YES. If it does not, code RING=NO (or omit the operand).

RING=YES is valid only for a switched line (DIAL=YES specified in the GROUP macro).

`[TADDR={character}]`
`       {NONE     }`

(tributary controller on BSC line only)

Specifies the station (or station and group) address you wish to assign to this communications controller if it is to represent a tributary station on a multipoint BSC line.

Code TADDR=character, where *character* is one or two EBCDIC characters specified in hexadecimal form (for example, C1 or C1C2). A one-character

address (C1) represents the station address. The second character in a two-character address (C1C2) represents the group address. Each address must be equivalent to the address that the access method sends to solicit input from the tributary controller.

If this communications controller does not represent a tributary station, code TADDR=NONE or omit the operand.

This operand is valid only for lines serviced by a type 2 scanner; the operand should not be specified if you have specified OPCSB2=YES in the BUILD macro and CHNPR1=HIGH in the LINE (or GROUP) macro for the line.

`[TERM=type]`

Specifies the type of station with which the program will communicate over the line represented by this LINE macro. It must be one of the types listed in Figure 3-7.

Note 1: If CUTYPE is coded on the LINE or GROUP macro, this operand is ignored.

Note 2: If different types of BSC devices (for example, 3275's and 3277's) are attached to the line, specify only one of the device types.

If type of station is:	*Code TERM=*
IBM 1030 Data Collection System	1030
IBM 1050 Data Communication System	1050
IBM 1060 Data Communication System	1060
IBM 1130 Computing System	1130
IBM 1800 Data Acquisition and Control System	1800
IBM System/360 Model 20	2020
IBM System/360 Model 25	2025
IBM 2260 Display Station	2260
IBM 2265 Display Station	2265
IBM 2701 Data Adapter Unit	2701
IBM 2703 Transmission Control	2703
IBM 2715 Transmission Control Unit Model 2	2715
IBM 2740 Model 1 Communications Terminal	2740-1
IBM 2740 Model 2 Communications Terminal	2740-2
IBM 2741 Communications Terminal	2741
IBM 2770 Data Communications System	2770
IBM 2780 Data Transmission Terminal	2780
IBM 2972 General Banking Terminal System:	
IBM 2980 Models 1 and 4 Teller Station	2980
IBM 2980 Model 2 Administrative Station	2980
IBM 3270 Information Display System:	
IBM 3275 Display Station	3275
IBM 3277 Display Station	3277
IBM 3284 Printer	3284
IBM 3286 Printer	3286
IBM 3650 Retail Store System (in BSC mode)	SYS3
IBM 3660 Supermarket System (in BSC mode)	SYS3
IBM 3670 Brokerage Communications System	3671
IBM 3704 Communications Controller	3704
IBM 3705 Communications Controller	3705
IBM 3735 Programmable Buffered Terminal	3735
IBM 3740 Data Entry System:	
IBM 3741 Data Stations	3741
IBM 3747 Data Converter	3747
IBM 3767 Communications Terminal (in start-stop mode)	
supported as 2740 Model 1	2740-1
supported as 2740 Model 2	2740-2
supported as 2741	2741
IBM 3770 Data Communications System (in BSC mode)	2770
IBM 3780 Data Transmission Terminal	3780
IBM 3790 Communications System	3790
IBM 3940 Banking Terminal	3940
IBM 3980 Banking Terminal	3980
IBM System/370 Model 125	3125
IBM System/370 Model 135	3135
IBM System/3	SYS3
IBM System/7 (BSC version)	SYS7
IBM System/7 (start-stop version)	2740-1
IBM System/32 (BSC version)	SYS3
IBM Communicating Magnetic Card Selectric ® Typewriter	2741
AT&T 83B3 Selective Calling Station	83B3
Western Union Plan 115A Outstation	115A
Western Union Teletypewriter Exchange Service	TWX
World Trade Teletypewriter Terminals	WTTY

Figure 3-7. Values for TERM Operand of LINE Macro

[UNITXC={YES}]
        {NO }

Specifies whether the emulation program is to signal Unit Exception status to the host processor when the program receives an EOT from the line.

It is normally appropriate to specify UNITXC=YES (or omit the operand), which causes the program to signal Unit Exception status upon receiving an EOT; however, if read and write commands within the access method are command chained, UNITXC=NO may be appropriate. UNITXC=NO, by suppressing the Unit Exception indication, prevents the command chain from being broken. (Unit Exception status always breaks the command chain.)

Note: Specify UNITXC=NO for 2741 terminals equipped with the break feature if you require CE and DE (Channel End and Device End) rather than the normal ending sequence (CE, DE, UE).

Generation Delimiter Macro Instruction (GENEND)

The GENEND macro indicates the end of the emulation program generation statements. It must be the last emulation program generation macro instruction coded.

The GENEND macro also specifies the scan limits and address substitution mask, if required, for each type 2 communication scanner installed in the communications controller and the scan limits and high speed select mask, if required, for each type 3 scanner installed. These parameters are for use only if any communication lines in the network operate at 4800 or more bits per second. Specifying these parameters causes the scanner to scan line interfaces to which high speed lines are attached more frequently than those for lower speed lines; the more frequent scanning is done at the expense of not scanning other line interface addresses. The addresses not scanned are therefore rendered unusable.

Use of scan limits, address substitution masks, and high speed select masks are described in more detail in Appendix I.

Name	*Operation*	*Operands*
[symbol]	GENEND	[HSPDSEL=([mask1],[mask2],[mask3],[mask4])] [,SCANCTL=([limit1],[limit2],[limit3],[limit4], [asmask])]

[symbol]

Is any symbol valid in the assembler language. It provides a name for the macro.

[HSPDSEL=([mask1],[mask2],[mask3],[mask4]) *(applicable to type 3 scanners only)*

Specifies the high speed select masks for each type 3 communication scanner installed in the communications controller. The masks are used to cause high speed line interfaces to be scanned more frequently than interfaces for lower speed lines (under 4800 bps).

mask1...mask4

Specifies eight-bit binary sequences (for example, 00101000) constituting the masks. For scanning purposes, the line interface bases (LIB) serviced by a type 3 scanner are divided into eight portions. The eight bit positions of a mask correspond to the eight portions (0-7) within all LIBs serviced by the scanner. See Appendix I for an illustration.

A mask bit of 0 specifies that all line interface addresses in the corresponding portion of the LIB are scanned equally often. A mask bit of 1 specifies that only the line interface with the lowest address within that LIB portion is scanned; all other addresses within that LIB portion are not scanned. The scans that would otherwise be applied to these addresses are instead applied to the lowest address, thus increasing the scan frequency of that address. See the table below for addresses scanned and not scanned for each high speed select mask bit position.

LIB Portion and HSS Mask Bit Position	*Bit Value*	*Scanner Position*	*Address Scanned*	*Addresses Not Scanned*
0	1	First	020	021,030,031,040,041
		Second	0A0	0A1,0B0,0B1,0C0,0C1,0D0,0D1
		Third	120	121,130,131,140,141,150,151
		Fourth	1A0	1A1,1B0,1B1,1C0,1C1,1D0,1D1
1	1	First	022	023,032,033,042,043
		Second	0A2	0A3,0B2,0B3,0C2,0C3,0D2,0D3
		Third	122	123,132,133,142,143,152,153
		Fourth	1A2	1A3,1B2,1B3,1C2,1C3,1D2,1D3
2	1	First	024	025,034,035,044,045
		Second	0A4	0A5,0B4,0B5,0C4,0C5,0D4,0D5
		Third	124	125,134,135,144,145,154,155
		Fourth	1A4	1A5,1B4,1B5,1C4,1C5,1D4,1D5
3	1	First	026	027,036,037,046,047
		Second	0A6	0A7,0B6,0B7,0C6,0C7,0D6,0D7
		Third	126	127,136,137,146,147,156,157
		Fourth	1A6	1A7,1B6,1B7,1C6,1C7,1D6,1D7
4	1	First	028	029,038,039,048,049
		Second	0A8	0A9,0B8,0B9,0C8,0C9,0D8,0D9
		Third	128	129,138,139,148,149,158,159
		Fourth	1A8	1A9,1B8,1B9,1C8,1C9,1D8,1D9
5	1	First	02A	02B,03A,03B,04A,04B
		Second	0AA	0AB,0BA,0BB,0CA,0CB,0DA,0DB
		Third	12A	12B,13A,13B,14A,14B,15A,15B
		Fourth	1AA	1AB,1BA,1BB,1CA,1CB,1DA,1DB
6	1	First	02C	02D,03C,03D,04C,04D
		Second	0AC	0AD,0BC,0BD,0CC,0CD,0DC,0DD
		Third	12C	12D,13C,13D,14C,14D,15C,15D
		Fourth	1AC	1AD,1BC,1BD,1CC,1CD,1DC,1DD
7	1	First	02E	02F,03E,03F,04E,04F
		Second	0AE	0AF,0BE,0BF,0CE,0CF,0DE,0DF
		Third	12E	12F,13E,13F,14E,14F,15E,15F
		Fourth	1AE	1AF,1BE,1BF,1CE,1CF,1DE,1DF
any	0	All addresses in corresponding scanner position are scanned.		

mask1 applies to a type 3 scanner installed in the first scanner position (base module), *mask2* to a type 3 scanner installed in the second scanner position (first expansion module), etc. If a scanner position does not contain a type 3 scanner, code a comma to represent the missing mask, if succeeding positions are occupied by a type 3 scanner.

The bit settings you specify should correspond to the high speed lines requiring increased scanning. For each such line interface installed in the controller, a high speed select feature is present that blocks the attachment of lines to all but the lowest address in the corresponding LIB portion. *Example:* Assume that a 3705 having three modules is equipped with type 3 scanners in the first and second expansion modules, but not in the base module. If high speed select features are present in the second scanner for LIB portions 3 and 7 (thus allowing high speed lines to be attached to addresses 0A6 and 0AE), you would specify HSPDSEL= (,00010001,00000000). The first comma signifies that no type 3 scanner is

installed in the base module; the first eight-bit mask indicates that increased scanning frequency is required for addresses 0A6 and 0AE in LIB portions 3 and 7, respectively; and the second mask indicates that no addresses in the second expansion module (scanner position 3) require increased scanning frequency.

If you omit the HSPDSEL operand but the program generation procedure determines that the high speed select function is required, the procedure determines the appropriate mask and assumes that the appropriate high speed select features are installed.

`[SCANCTL=([limit1],[limit2],[limit3],[limit4],asmask)]` *(applicable to type 2 and 3 scanner only)*

Specifies the scan limits for each type 2 and type 3 communication scanner installed in the controller and specifies the address substitution mask, if used.

This operand is valid only if one or more type 2 and type 3 communication scanners are installed in a 3705 or in a 3704 equipped with the communication scanner expansion feature. (An address substitution mask must not be specified if a type 3 scanner is installed.)

Omit this operand if the controller is equipped with a type 1 scanner.

For a 3704 equipped with a type 2 scanner: (1) do not specify an address substitution mask; (2) specify only one scan limit—SCANCTL=1 or SCANCTL=3; SCANCTL=2 is not valid.

`limit1...limit4`

Specifies the scan limits for each installed type 2 or type 3 scanner. Each limit can be from 0 to 3; these values have the meanings shown below. *Limit1* specifies the scan limit for the first scanner position (base module), *limit2* for the second position (first expansion module), etc. All addresses associated with a scanner are scanned if the scan limit for that scanner is 0. Scan limits of 1, 2, and 3 reduct the number of addresses scanned to 8, 48, and 16, respectively. If a scanner position does not contain a type 2 or type 3 scanner, code a comma for the corresponding limit [for example, SCANCTL=(limit1,,limit2,,asmask)]. If a type 2 or type 3 scanner is installed but you specify no limit, the generation procedure assigns the appropriate limit based on the range of actual installed addresses and line speeds as specified in the LINE macros.

The scan limits have the following meanings:

Scan Limit	*Addresses Scanned*	*Addresses Not Scanned*	*Maximum Line Speed*
For IBM 3705:			
0	020-05F 0A0-0FF 120-17F 1A0-1FF	(all addresses scanned)	4,800 bps
1	020-027 0A0-0A7 120-127 1A0-1A7	028-05F 0A8-0FF 128-17F 1A8-1FF	56,000 bps
2	020-04F 0A0-0CF 120-14F 1A0-1CF	050-05F 0D0-0FF 150-17F 1D0-1FF	9,600 bps
3	020-02F 0A0-0AF 120-12F 1A0-1AF	030-05F 0B0-0FF 130-17F 1B0-1FF	19,200 bps
For IBM 3704:			
0	020-03F	(all addresses scanned)	4,800 bps
1	020-027	028-03F	50,000 bps
2	020-03F	(all addresses scanned)	9,600 bps
3	020-02F	030-03F	19,200 bps

`asmask`

Specifies the address substitution mask to be used if the communications controller is equipped with the address substitution feature. Specify the mask as a binary sequence of four bits (omitting frame characters, B'') as follows:

Bit	*Value*	*Meaning*
0	1	Address substitution is to be performed for address 0 in LIB position 1. Addresses E and F in all LIB positions are disabled.
0	0	No address substitution; all addresses enabled.
1	1	Address substitution is to be performed for address 2 in LIB position 1. Addresses C and D in all LIB positions are disabled.
1	0	No address substitution; all addresses enabled.
2	1	Address substitution is to be performed for address 4 in LIB position 1. Address A and B in all LIB positions are disabled.
2	0	No address substitution; all addresses enabled.
3	1	Address substitution is to be performed for address 6 in LIB position 1. Addresses 8 and 9 in all LIB positions are disabled.
3	0	No address substitution; all addresses enabled.

Caution: The address substitution mask should not be specified if one or more type 3 scanners are installed in the communications controller because address substitution inhibits scanning of corresponding addresses in *all* LIBs regardless of whether serviced by type 2 or type 3 scanners. Instead of address substitution use upper scan limits or high speed select masks to provide increased scanning frequency for high-speed links.

If you omit the SCANCTL operand, the generation procedure automatically calculates the appropriate scan limits, and, if the network configuration requires the use of address substitution, calculates the address substitution mask. The procedure assumes that the appropriate Address Substitution feature is installed. A message is printed in the assembly listing when the feature is required. Determine from the system designer whether the feature is installed. If not, a discrepancy exists; either respecify the network configuration or have the Address Substitution feature installed.

Relationship Between Program Generation Macros and Modules

The following identifies all emulation program modules that can be included by stage 2 of emulation program generation and indicates the conditions for inclusion:

Modules Included for Type 1 Channel Adapter

Module Name	*Conditions for Inclusion*
One of the following two CYANUC modules is included.	
CYANUC10	TYPE=TYPE1 in a CSB macro.
CYANUC20	TYPE=TYPE2 in a CSB macro.
One of the following two CYASVC modules is included.	
CYASVC10	TYPE=TYPE1 in a CSB macro.
CYASVC20	TYPE=TYPE2 in a CSB macro.
No more than one of the following two CYASIS modules is included.	
CYASIS10	TYPE=TYPE1 in a CSB macro *and* LNCTL=SS in a GROUP macro.
CYASIS20	TYPE=TYPE2 in a CSB macro *and* LNCTL=SS in a GROUP macro.
No more than one of the following two CYABIS modules is included.	
CYABIS10	TYPE=TYPE1 in a CSB macro *and* LNCTL=BSC in a GROUP macro.
CYABIS20	TYPE=TYPE2 in a CSB macro *and* LNCTL=BSC in a GROUP macro.
No more than one of the following eight CYASL modules is included.	
CYASL110	TYPE=TYPE1 in a CSB macro *and* LNCTL=SS in a GROUP macro, *and* TERM=1030, 1050, 1060, 2740-1, 2740-2, 2741, or SYS7 in a LINE or GROUP macro.
CYASL210	TYPE=TYPE1 in a CSB macro *and* LNCTL=SS in a GROUP macro, *and* TERM=115A, 83B3, TWX, or WTTY in a LINE or GROUP macro.
CYASL310	TYPE=TYPE1 in a CSB macro *and* LNCTL=SS in a GROUP macro, *and* TERM=1030, 1050, 1060, 2740-1 2740-2, 2741, or SYS7 *and* TERM=115A, 83B3, TWX, or WTTY in a LINE or GROUP macro.
CYASL410	TYPE=TYPE1 in a CSB macro *and* CU=2848 or 2845 or TERM=2260 or 2265 in a GROUP or LINE macro *and* neither CYASL110, CYASL210, nor CYASL310 is included in the program.
CYASL120	TYPE=TYPE2 in a CSB macro *and* LNCTL=SS in a GROUP macro *and* TERM=1030, 1050, 1060, 2740-1, 2740-2, 2741, or SYS7 in a LINE or GROUP macro.
CYASL220	TYPE=TYPE2 in a CSB macro *and* LNCTL=SS in a GROUP macro *and* TERM=115A, 83B3, TWX or WTTY in a LINE or GROUP macro.

Module Name	*Conditions for Inclusion*
CYASL320	TYPE=TYPE2 in a CSB macro *and* LNCTL=SS in in a GROUP macro *and* TERM=1030, 1050, 1060, 2740-1, 2740-2, 2741 or SYS7 *and* TERM=115A, 83B3, TWX, or WTTY in LINE or GROUP macros.
CYASL420	TYPE=TYPE2 in a CSB macro *and* CU=2848 or 2845 or TERM=2260 or 2265 in a GROUP or LINE macro *and* neither CYASL120, CYASL220, nor CYASL320 is included in the program.

No more than one of the following CYABL modules is included:

CYABL110	TYPE=TYPE1 in a CSB macro *and* LNCTL=BSC in a GROUP macro *and* CODE=EBCDIC in a LINE or GROUP macro.
CYABL210	TYPE=TYPE1 in a CSB macro *and* LNCTL=BSC in a GROUP macro *and* CODE=USASCII in a LINE or GROUP macro.
CYABL310	TYPE=TYPE1 in a CSB macro *and* LNCTL=BSC in a GROUP macro *and* CODE=EBCDIC and CODE=USASCII or FEATURE=DUALCODE in a LINE or GROUP macro.
CYABL120	TYPE=TYPE2 in a CSB macro *and* LNCTL=BSC in a GROUP macro and CODE=EBCDIC in a LINE or GROUP macro.
CYABL220	TYPE=TYPE2 in a CSB macro *and* LNCTL=BSC in a GROUP macro *and* CODE=USASCII in a LINE or GROUP macro.
CYABL320	TYPE=TYPE2 in a CSB macro *and* LNCTL=BSC in a GROUP macro *and* CODE=EBCDIC and CODE=USASCII or FEATURE=DUALCODE in a LINE or GROUP macro.

No more than one of the following CYADSP modules is included:

CYADSP10	TYPE=TYPE1 in a CSB macro *and* CUTYPE=2848 in a LINE or GROUP macro.
CYADSP20	TYPE=TYPE2 in a CSB macro *and* CUTYPE=2848 in a LINE or GROUP macro.

No more than one of the following CYABIT modules is included:

CYABIT10	TYPE=TYPE1 in a CSB macro *and* LNCTL=SS in a GROUP macro.
CYABIT20	TYPE=TYPE1 in a CSB macro *and* LNCTL=BSC in a GROUP macro.
CYABIT30	TYPE=TYPE1 in a CSB macro *and* LNCTL=SS *and* LNCTL=BSC in GROUP macros.

No more than one of the following CYATST modules is included:

CYATST10	TYPE=TYPE1 in a CSB macro *and* TEST=YES in the BUILD macro.
CYATST11	TYPE=TYPE1 in a CSB macro *and* TEST=YES in the BUILD macro *and* AUTO=lineaddr in a LINE macro.

Module Name	*Conditions for Inclusion*
CYATST20	TYPE=TYPE2 in a CSB macro *and* TEST=YES in the BUILD macro.
CYATST21	TYPE=TYPE2 in a CSB macro *and* TEST=YES in the BUILD macro *and* AUTO=line addr on a LINE macro.

No more than one of the following CYATRC modules is included:

CYATRC10	TYPE=TYPE1 in a CSB macro *and* LINETRC=YES in the BUILD macro.
CYATRC20	TYPE=TYPE2 in a CSB macro *and* LINETRC=YES in the BUILD macro.

No more than one of the following CYADAT modules in included:

CYADAT10	TYPE=TYPE1 in a CSB macro *and* DELAY=600 or DELAY=1200 on the GROUP macro.
CYADAT20	TYPE=TYPE2 on a CSB macro *and* DELAY=600 or DELAY=1200 on the GROUP macro.

The following module in included under the conditions shown:

CYADSS	DYNADMP=YES in the BUILD macro.

Modules Included for Type 4 Channel Adapter

Module Name	*Condition for Inclusion*

The following module is always included:

CYENUC

The following modules are included under the conditions shown:

CYESIS	LNCTL=SS in a GROUP macro.
CYEBIS	LNCTL=BSC in a GROUP macro.

No more than one of the following modules is included:

CYESL1	LNCTL=SS in a GROUP macro *and* TERM=1030, 1050, 1060, 2740-1, 2740-2, 2741, or SYS7 in a LINE or GROUP macro.
CYESL2	LNCTL=SS in a GROUP macro *and* TERM=83B3, 115A, TWX, or WTTY in a LINE or GROUP macro.
CYESL3	LNCTL=SS in a GROUP macro *and* TERM=1030, 1050, 1060, 2740-1, 2740-2, 2741, or SYS7 in a LINE or GROUP macro *and* TERM=83B3, 115A, TWX, or WTTY in a LINE or GROUP macro.
CYESL4	CU=2848 or 2845 or TERM=2260 or 2265 in a LINE or GROUP macro *and* neither CYESL1, CYESL2, or CYESL3 is included in the program.

No more than one of the following modules is included:

CYEBL1	LNCTL=BSC in a GROUP macro *and* CODE=EBCDIC in a LINE or GROUP macro.
CYEBL2	LNCTL=BSC in a GROUP macro *and* CODE=USASCII in a LINE or GROUP macro.

Module Name	*Condition for Inclusion*
CYEBL3	LNCTL=BSC in a CSB macro *and* CODE=EBCDIC *and* CODE=USASCII or FEATURE=DUALCODE in a LINE or GROUP macro.

No more than one of the following modules is included:

CYETST2	TYPE=TYPE2 in a CSB macro *and* TEST=YES in the BUILD macro.
CYETST3	TYPE=TYPE3 in a CSB macro *and* TEST=YES in the BUILD macro.
CYETST4	TYPE=TYPE2 *and* TYPE=TYPE3 in at least one CSB macro each *and* TEST=YES in the BUILD macro.

One of the following SVC modules is included:

CYESVC	LNCTL operand of any GROUP macro does not specify ALC.
CYESVCA	LNCTL operand of a GROUP macro specifies ALC (LNCTL=ALC).

One of the following CDS modules is included:

CYECDS	LNCTL operand of any GROUP macro does not specify ALC.
CYECDSA	LNCTL operand of a GROUP macro specifies ALC (LNCTL=ALC)

The following modules are included under the conditions shown:

CYEALC	LNCTL=ALC in a GROUP macro.
CYEBIS	LNCTL=BSC in a GROUP macro.
CYEBLW	TYPE=TYPE3 in a CSB macro.
CYEDAT	DELAY=600 or 1200 or QUIETCT does not equal 0 in a GROUP macro.
CYEDSP	CUTYPE=2848 in a LINE or GROUP macro.
CYEDSS	DYNADMP=YES in the BUILD macro.
CYESIS	LNCTL=SS in a GROUP macro.
CYETRC	LINETRC=YES in the BUILD macro.

Chapter 4: Generating an Emulation Program under OS

The emulation program generation procedure under OS is a two-stage process consisting of a series of jobs executed under the control of the operating system. You must code the entire stage one input job stream. The stage two job stream is produced automatically by stage one and therefore does not require coding.

Stage one of the generation procedure is a series of assembly steps using either the communications controller assembler or an OS assembler to prepare from the emulation program generation macros (describing the emulation program to be generated), a job stream (sequential data set) for input to stage two. This job stream can be directed to cards, tape, or a direct access storage unit. The stage one output (stage two input) contains (1) data constants, (2) macros that will cause stage two to generate the control tables and conditionally assemble the required program modules, (3) job control statements for stage two and (4) linkage editor control statements (the F-level linkage editor is required).

Operator intervention is required between the stages of program generation. Diagnostic messages produced at the end of stage one indicate any errors that have occurred. If these are serious errors, an incomplete job stream is produced. The source statements must be corrected and stage one must be re-executed. If no serious errors occur in stage one, the operator initiates the second stage, specifying as input the stage one output. Refer to Appendix D for the diagnostic messages that may appear in the stage one output listing.

Stage two of the generation procedure first uses the operating system assembler to assemble the control tables and those program modules that require conditional assembly, and places the resultant object modules on the library you have specified in the OBJLIB operand of the BUILD macro. Stage two then link edits these modules and other, preassembled, modules (located in SYS1.OBJ3705) into an EP load module and places this module on the library you have specified in the LOADLIB operand of the BUILD macro. From this library the loader provided in the system support programs may load the emulation program into the communications controller.

In addition to the load module produced by the linkage editor, unresolved external references may also be produced. See the section "Unresolved External References" in this chapter for additional information.

Upon the completion of stage two, the load module is ready to be loaded into the communications controller from the host processor.

Note: Multiple emulation program load modules must have different names or the latest copy will destroy the previous copy if same library is used for both.

Figure 4-1 shows the contents of the stage one input job stream using the OS assembler. For an example of the contents of the stage one input job stream using the communications controller assembler, refer to "Storage Requirements and Job Control Language" in the appendix section of the *Assembler Language* manual (see Preface). Figure 4-2 shows the contents of the stage two input job stream produced automatically by stage one.

```
//STAGE1   JOB        MSGLEVEL=1, ...
//STEP1    EXEC       ASMFC,PARM='DECK'
//SYSLIB   DD         DSN=SYS1.MAC3705...
           .
           .
           .
//SYSIN    DD         *
           .
           .
           .
           BUILD      (Source program
           .          generation
           .          statements)
           .
           GENEND
           END
/*
//
```

Figure 4-1. OS Generation Stage One Input Job Stream

```
//STAGE2   JOB        MSGLEVEL=1, ...
//STEP1    EXEC       PGM=IFKASM
//SYSLIB   DD         DSNAME=SYS1.MAC3705
           .
           .
           .
  (JCL for communications controller assembler)
           .
           .
           .
//SYSIN    DD         *
  (Data for conditional assembly of CYAEPCCB, CYAEPLGT,
  CYACHVT, and CYALNVT modules placed on OBJLIB)
           .
           .
           .
/*
//STEP2    EXEC       PGM=IEWL,REGION=128K
           .
           .
           .
  (JCL statements for F-level linkage editor)
           .
           .
           .
//SYSLIN   DD         *
           .
           .
           .
  (Data for partial linkage editing)
           .
           .
           .
/*
//STEP3    EXEC       PGM=IEWL,REGION=128K
           .
           .
           .
  (JCL statements for F-level linkage editor)
//SYSLIN   DD         *
           .
           .
           .
  Data for link-editing consists of INCLUDE cards for modules
  from SYS1.OBJ3705 plus modules from the library
  specified in the OBJLIB operand of the BUILD macro.
  The EP load module is placed in the library specified in
  the LOADLIB operand of the BUILD macro.
/*
//
```

Figure 4-2. OS Generation Stage Two Input Job Stream

Reminder: If the load module requested on the EXEC statement resides in a private library, the system must be informed of that fact. Unless the system is told that the load module requested on the EXEC statement resides in a private library, the system expects to find the program in the system library (SYS1.LINKLIB). One way to tell the system that the load module that the stage two job stream needs resides in a private library is to include a STEPLIB DD statement as one of the DD statements for that step. The STEPLIB card should follow the card defined as //STEP1.

Unresolved External References

If unresolved external references occur in the listing and meet the conditions listed below, they should be ignored.

	Unresolved references for:	
	Type 1 Channel Adapter	*Type 4 Channel Adapter*
If trace option is not specified:	CYATRCEI	CYATRCEI CYETRCRS CYETRCSP
If trace option is not specified and dynamic dump option is specified:	CYASETRC CYATABLE	CYASETRC CYATABLE
If trace option is not specified and BSC terminals are specified:		CYATRCL2 CYETRCL2
If dynamic dump option is specified:	$DSCCB	
If dynamic dump option is not specified:	CYADSTRT	CYADSTRT
If panel test option is not specified:	CYAPANLT	CYAPANLT
(type 1 scanner without Autocall)	CYATST10	
(type 1 scanner with Autocall)	CYATST11	
(type2/3 scanner without Autocall)	CYATST20	
(type 2/3 scanner with Autocall)	CYATST21	
If panel test option is specified and no EBCDIC lines are serviced by a Type 2 scanner:		CYAEBCDT
If panel test option is specified and no ASCII lines are serviced by a Type 2 scanner:		CYAASCDT

	Unresolved references for: *Type 1 Channel Adapter*	*Type 4 Channel Adapter*
If panel test option is specified and no BSC terminals are specified:		CYAEBCDT CYAASCDT
If panel test option is specified and no start-stop terminals are specified:		CYAXTABL
If start-stop terminals are specified but no display terminals are specified:	CYAATDA4 CYAB28CL CYAB2848	CYAATDA4 CYAATDA5 CYAB28CL CYAB2848
If start-stop terminals are specified but no teletypewriter (83B3, 115A) or TWX terminals are specified:	CYATDONE	CYATSTYE CYASRCH
If start-stop terminals are specified but no DELAY or QUIETCT operand is specified:	CYADAT1 CYADAT2	CYADAT1 CYADAT2
If no start-stop terminals are specified:	CYABARP1 CYABTDA0 CYACBKPL CYACBRES CYACPOLS CYACPRES CYACRDCL CYACREAS CYACSEAS CYACWRIS CYAMTBFR CYAQUIET *CYASPCFA *CYASPCFB *CYASPCF8 *CYASRCVT CYASTPER CYATRN *CYAXSSTT	CYABARP1 CYABTDA0 CYACBKPL CYACBRES CYACPOLS CYACPRES CYACRDCL CYACREAS CYACSEAS CYACWRIS CYAMTBFR CYAQUIET CYASTPER CYATDONE CYATRN CYATSTYE

*These modules are unresolved only if a type 1 scanner is specified

If BSC terminals are specified:

If no type 2 scanner is specified:	CYARARS0 CYATAPD0 CYATAX10 CYATBSWR CYATSTMW CYATXDA0
If no type 3 scanner is specified:	CYEABRTW CYEPRPRC CYERCVN CYERCVN1 CYERCVP CYERCVPS CYERCVP1 CYETXEND CYEXITB CYEXMITN

	Unresolved references for:	
	Type 1 Channel Adapter	*Type 4 Channel Adapter*
		CYEXMSYN
		CYEXMTEN
		CYEXPOLL
		CYEXTEND
If no BSC terminals are specified:		
	* CYABPCFA	CYABSHIO
	* CYABPCF8	CYABSTOP
	CYABSHIO	CYACADPB
	CYABSTOP	CYACPOLB
	CYACADPB	CYACPREB
	CYACPOLB	CYACREAB
	CYACPREB	CYACSEAB
	CYACREAB	CYACSETB
	CYACSEAB	CYACWRIB
	CYACSETB	CYARARS0
	CYACWRIB	CYATAPD0
	* CYAPCF45	CYATBSPL
	CYARARS0	CYATBSPR
	* CYARCDTA	CYATBSRD
	CYATAPD0	CYATBSSM
	CYATBSPL	CYATBSWR
*These modules are unresolved only if a type 1 scanner is specified	CYATBSPR	CYATSTMW
	CYATBSRD	CYAPSUDO
	CYATBSSM	CYARCVPS
	CATBSWR	CYERPLTO
	CYATSTMW	CYETMPHS
		CYETOTRN
		CYETXEND
		CYETXWTO
		CYEXITB
		CYEXMITO
		CYEXMSYN
		CYEXMTEN
		CYEXPOLL
		CYEXTEND
		CYEXWTRN
If no ALC lines (LNCTL=ALC) are specified:		CYETMALP
		CYETMALR
		CYETMALW

Notes: (1) External reference TM598 is defined in the CYASL210/310/320 modules. (2) External reference CYAATDA5 is defined in the CYADSP10/20 modules.

Chapter 5: Emulation Program Utilities under OS

This chapter explains the use of the loader, the dump, and the dynamic dump utilities under the control of the Operating System (OS). In order to communicate with the communications controller, the loader, dump, and dynamic dump utility programs require the existence of a native subchannel unit control block (UCB). A 3705 unit control block can be generated into any system after and including release 21.7 of the Operating System. A 3704 unit control block can be generated into any system after and including release 21.8 of the Operating System. For earlier releases, an unused UCB with the appropriate address must be modified to describe the controller as follows: change UCBETI to X'00', UCBATI to X'00', and UCBTYP to X'50004015'.

If a UCB with the appropriate address does not exist, an I/O system generation may be performed to include it. The above modification may then be performed. The UCB being modified must not represent a device in a unitname class that would cause it to be allocated without specifically referring to it.

The Loader Utility

The loader utility program transfers an emulation program load module from the host processor to the communications controller. It must be run as a job or job step under OS.

The loader has two modules. One is an operating system utility that may be invoked as any other OS utility. The other module runs in the communications controller. When the loader is invoked, the controller module is contained within a data area in the host processor loader module. The host processor module loads the controller module into the controller via an initial program load (IPL) command.

The communications controller module of the loader can be executed in any communications controller. The only requirements for the load operation are that the communications controller be identified to the operating system, that it be free to be allocated to the loader job step, and that its power be on. See Appendix C for host storage requirements.

Before the loader utility loads the emulation program into the controller, it loads a diagnostic routine, called the *initial test* routine. This routine tests the communications controller for hardware malfunctions that might later cause failure of the emulation program and sets good parity in EP storage areas. If the initial test routine detects no malfunctions, the loader then loads the emulation program into the controller. If the initial test routine does detect trouble, that routine stops and the loader issues error message IFL005I indicating the fact.

Loading the initial test routine is optional (it is run unless you specify its omission in the LOAD control statement), but is recommended because it can detect conditions that can later cause failure of the emulation program. Running the initial test routine is especially recommended for a communications controller which has just been powered on because it will set good parity in storage.

Successful completion of the emulation program loading process is indicated to the CPU operator by a write-to-operator message.

Syntax errors in the LOAD statement or permanent I/O errors occurring during loading are indicated by messages sent to the message data set (SYSPRINT).

Messages issued by the loader are given in Appendix E.

Input to the Loader Program

Either two or three data sets are used as input to the loader:

- A DASD partitioned data set (input data set) containing the emulation program load module to be loaded.
- A data set containing the LOAD control statement specifying the name of the emulation program load module and the communications controller into which the module is to be loaded.
- A partitioned data set containing the initial test routine (consisting of load modules IFL3705D and IFL3705E) to be loaded before emulation program loading. This data set is optional and may be omitted if the initial test routine is not desired (as indicated by DIAG=NO in the LOAD statement).

The loader consists of the load modules IFLOADRN, IFLLD1P2, and IFLLD2P2. These modules must be in the SYS1.LINKLIB data set or on a partitioned data set pointed to by a STEPLIB or JOBLIB statement. If the initial test routine is desired, the loader must also consist of IFL3705A and IFL3705B.

Output from the Loader Program

The loader produces one output data set (SYSPRINT). This data set contains the completion messages and/or error messages produced by the loader. See Appendix E for the loader utility messages.

Job Control Statements

The OS job control statements needed to invoke the loader program are as follows.

`//jobname`	JOB	(initiates the job)
`//`	EXEC	(specifies the program name [IFLOADRN], or the name of a procedure that contains the JCL)
`//SYSUT1`	DD	(specifies the DASD input data set that contains the emulation program load module)
`//SYSPRINT`	DD	(specifies the message data set)
`//SYSUT3`	DD	(specifies the DASD input data set that contains the communications controller initial test routine [load modules IFL3705D and IFL3705E]. SYSUT3 is not required if DIAG=NO is specified on the Loader control statement.)
`//ccuname`	DD	(specifies the unit address of the communications controller to be loaded)
`//SYSIN`	DD	(specifies the data set [input stream] that contains the utility control statement(s))

Utility Control Statement (LOAD)

There is one utility control statement: the LOAD statement. This statement may not be omitted and must be contained on an 80-character card image (the loader program does not recognize continuation characters). The LOAD statement

specifies (1) which member of the input data set contains the emulation program load module to be loaded, (2) which communications controller is to be loaded, and (3) whether or not the diagnostic initial test routine is to be executed before the emulation program load module is loaded.

Name	*Operation*	*Operands*
	LOAD	LOADMOD=member name, 3705=ccuddname [,DIAG={Y6}] {Y8} {NO}

LOADMOD=member name

Specifies which member of the input data set indicated by SYSUT1 contains the desired emulation program load module. This member must be in standard OS load module form, with the 'DC' link-edit parameter specified without the 'overlay' or 'sctr' (scatter) parameters.

3705=ccuddname

Specifies the ccuname given to the DD statement that identifies the communications controller to be loaded.

DIAG={Y6}
{Y8}
{NO}

Specifies whether the loader is to load the initial test routine into a 3704 or a 3705 without extended addressing (DIAG=Y6), a 3705 with extended addressing (DIAG=Y8), or is not to load the routine at all (DIAG=NO).

Example of Job and Utility Control Statements

Assume that an emulation program load module named ECP1 resides on a data set named ALLEPS. This module is to be loaded into a communications controller, the unit address of which is 030. The control and utility statements would be similar to:

```
//CCULOAD     JOB     MSGLEVEL=(1,1)
//            EXEC    PGM=IFLOADRN
//SYSUT1      DD      DSNAME=ALLEPS,UNIT=2311,          X
//                    VOL=SER=111111,DISP=OLD
//SYSUT3      DD      DSNAME=SYS1.LINKLIB,UNIT=2311,    X
//                    DISP=SHR
//SYSPRINT    DD      SYSOUT=A
//CCU030      DD      UNIT=030
//SYSIN       DD      *
            LOAD    LOADMOD=ECP1,3705=CCU030
/*
```

This example assumes that the initial test routine (located on SYS1.LINKLIB) is to be loaded and executed before the emulation program is loaded. If the initial test is not wanted, you would specify DIAG=NO in the LOAD statement (or omit the DIAG parameter) and omit the SYSUT2 DD statement.

The Dump Utility

The dump utility program is used to dump the storage contents of a 3704 or 3705 communications controller. It accomplishes this by a two-step process:

Step 1: The storage contents of the controller are copied to a direct-access data set (SYSUT2).

Step 2: A printable copy of the controller's storage contents is produced and placed on a sequential output data set (SYSPRINT). The SYSUT2 data set from step one serves as input to this step.

The dump program also allows you to print the results of a TCAM dump if your TCAM program has DMPAUTO=YES specified.

Executing the dump utility stops operation of the emulation program. Once the dump utility has been started, it should not be canceled before normal completion. If the communications controller contains an active emulation program, the host module writes a message to the operator and waits for a response to verify that the operator wishes to dump the controller. If he does not, he enters a response that ends the dump job.

Refer to Appendix C for the storage requirements of the dump utility.

Dumping the Controller Storage

Dumping from the controller to the direct-access data set is performed by the first step of the dump utility. This step first transfers into the communications controller a module containing the utility code needed for the controller to participate in the dumping process. (This module is contained within the dump program in the host processor until transferred to the controller via an initial program load (IPL) command.)

Step one always transfers the entire contents of controller storage and local store registers to the host processor, which places them on a direct access data set. However, a small portion of the storage data is overlaid by the dumping process (by either read-only storage [ROS] or dump utility code.) The areas not available are as follows:

For a 3705 equipped with a single channel adapter:

Hexadecimal addresses:	Overlaid by:
0 through 1FF	ROS
400 through 4E7	dump utility code
700 through 707	ROS
780 through 79F	ROS

For a 3705 equipped with two channel adapters or a 3704:

Hexadecimal addresses:	Overlaid by:
0 through 3FF	ROS
400 through 4E7	dump utility code
700 through 70F	ROS
780 through 79F	ROS

Of the overlaid areas described above, locations 0 through X'1FF' are not printed.

Note: The contents of the controller's external registers are not transferred to the host processor. If the contents of these registers must be examined, they must be displayed on the controller's operator panel and the contents noted before the dump utility is invoked.

When step one is complete, the program informs the CPU operator. At this point the controller is idle and can be reloaded with an emulation program via the loader utility.

For the job control statements needed to both dump and print the contents of controller storage, see the topic, "How to Dump and Print Storage Contents" later in this chapter.

Printing the Dump Data

The second step of the dump utility converts all or a selected part of the dumped data to printable form, then places the data on a sequential output data set. The output listing shows the hexadecimal representation of controller storage and register contents, and gives the character equivalents of all EBCDIC bit patterns that represent characters. Beyond this, two options are available, as specified by the DUMP control statement:

- Formatted or unformatted dump of controller storage. The unformatted storage dump is printed with EBCDIC equivalents on the right side of the page. In a formatted storage dump, the mnemonic operation codes are printed with the instructions.
- The complete contents of storage may be listed, or any specified portion or portions of storage.

Utility Control Statement (DUMP)

The dump utility program requires one control statement, DUMP. This statement specifies that the dump listing shows either the entire contents of communications controller storage or a portion thereof.

Name	*Operation*	*Operands*
[symbol]	DUMP	[FROMADDR=address] [,TOADDR=address] [,MNEMONIC={Y}] {N}

[symbol]

Specifies a name, one to eight characters in length, beginning with an alphabetic character.

FROMADDR=address

Specifies the lower limit of the communications controller storage to appear on the listing. If you omit FROMADDR, the listing starts at X'200'. (If you specify a value less than X'200', error message IFL201I is issued and a dump of the entire storage contents is produced).

TOADDR=address

Specifies the upper limit of communications controller storage to appear on the dump listing. If you omit TOADDR, the listing ends at the upper limit of communications controller storage. (If you specify a value higher than the upper limit of storage, message IFL201I is issued and a dump of the entire storage contents is produced).

MNEMONIC={Y}
{N}

Specifies whether or not the IFLDUMP program is to print the mnemonic operation codes in the dump listing. These codes are printed only if you specify MNEMONIC=Y. When MNEMONIC=Y is specified, the paper output is approximately three times greater than when MNEMONIC=N is specified.

Note: If the controller storage to be dumped contains line trace information, omission of mnemonic operation codes is recommended.

How to Dump and Print Storage Contents

To use the dump utility to both dump and print the controller storage contents, you provide job control statements only for the first step and the DUMP control statement(s) for the second step. The first step will invoke the second step when the first step is completed. The statements are:

//jobname	JOB	(initiates the job)
//STEP1	EXEC	(specifies the program IFLREAD, or the name of a procedure containing the job control statements)
//SYSUT1	DD	(specifies the communications controller whose contents are to be dumped)
//SYSUT2	DD	(specifies the temporary DASD work data set onto which the contents of the communications controller are to be dumped)
//SYSPRINT	DD	(specifies a sequential data set [system output device, magnetic tape, or DASD volume] onto which the dump program is to place the dump listing— the output block size is 3509)
//SYSIN	DD	(specifies the data set [input stream] that contains the utility control statement, DUMP)

Assume, for example, that a communications controller whose unit address is 030 is to be dumped, that the dump listing is to show the contents of communications controller storage from address X'17F0' to the end, and that mnemonic operation codes are not to be printed.

The control and utility statements would be similar to:

```
//CCUDUMP      JOB     MSGLEVEL=(1,1)
//STEP2        EXEC    PGM=IFLREAD
//SYSUT1       DD      UNIT=030
//SYSUT2       DD      UNIT=SYSDA,SPACE=(512,(481),,CONTIG),   X
//                     DISP=NEW
//SYSPRINT     DD      SYSOUT=A
//SYSIN        DD      *
                DUMP    FROMADDR=17F0
/*
//
```

Under TCAM (Telecommunications Access Method), the communications controller may be dumped to a dump data set; however, TCAM will not print this data set. To print the data set you must run step 2 of the dump utility described above. The job control language required to accomplish this is similar to that given in step 2 of the example. The only differences are:

- 'EXEC IFLDUMP' replaces 'EXEC IFLREAD'.
- The SYSUT1 card is not required.
- SYSUT2 must refer to the same data set filled by TCAM's 3705 dump command.

The Dynamic Dump Utility

The dynamic dump utility is an optional utility that provides the following services that are useful in debugging. This utility can be used to:

- Obtain, without terminating the execution of the emulation program, (1) a storage dump (from location 0 through the end of storage) of the communications controller, or (2) a display, on the operator's console at the host processor, of portions of controller storage (up to 144 bytes) starting at any location, or (3) a dump of the emulation program trace table only.
- Activate or deactivate the emulation program line trace function.
- Obtain a dynamic dump of emulation program trace table entries as they are entered into the trace table.

Input to the Dynamic Dump Utility

Control statements are used to request the various functions of the dynamic dump utility. These control statements may reside in the SYSIN data set (input stream) or they may be entered via the operator's console.

Initially, the dynamic dump utility reads control statements from the SYSIN data set until either an END statement or a PAUSE statement is read. The PAUSE statement instructs the dynamic dump utility to read control statements only from the operator's console until either an END statement or a SYSIN statement is read. The SYSIN statement is the opposite of the PAUSE statement; it instructs the dynamic dump utility to return to the SYSIN data set for control statements (beginning with the next statement after the last PAUSE statement). An END statement either encountered in the SYSIN data set or entered from the operator's console causes the dynamic dump utility to be terminated.

Output from the Dynamic Dump Utility

Work Data Set—This is a temporary data set on which the contents of storage are written. (This data set usually resides on a tape unit.)

Output Data Set—This is the data set on which the trace or storage dump is printed from the work data set. It also contains the dynamic dump control statements and applicable error messages.

Operator's Console—The operator's console of the host processor may receive output as a result of a DISPLAY statement, control statement responses, and error conditions.

Dynamic Dump Operational Characteristics

The dynamic dump utility is used when trouble or error conditions indicate that a dynamic dump of controller storage is desirable to isolate and fix a problem.

The dynamic dump utility physically consists of two modules. One module resides in the host processor (as load module IFLSVEP), and the other module resides in the controller as part of the emulation program. (This module is included in the emulation program only if DYNADMP=YES is specified in the BUILD macro during emulation program generation.) These two modules communicate with each other to transfer specified controller storage to the host module. If a DISPLAY command is used to enter a request, the transferred storage is displayed at the operator's console or sent to the output data set; otherwise, the host module will write the received storage to the work data set in 516 byte blocks. You may

then invoke the PRINT facility of the dynamic dump utility to print the contents of this work data set.

When a particular user request has been satisfied, the host module of the dynamic dump utility issues message IFL503I to inform the operator that the transfer of data to the work data set is complete.

Obtaining a Dynamic Dump of Trace Table Entries

The most important function of the dynamic dump utility is its ability to dynamically dump emulation program trace entries. The following paragraphs describe this operation. Refer to "Utility Control Statements" in this chapter for additional information on the control statements discussed and to the "Example of Dynamically Dumping Trace Table Entries," later in this section, for an illustration of the input stream.

To dynamically dump the emulation program trace table entries, first start the trace on the desired range of emulation program subchannels by using the OPTION control statement or by using the communications controller control panel. To initiate the transfer of 516-byte blocks of trace entries (hereafter referred to as *trace blocks*) to the dynamic dump utility host module, enter the DY Dynamic control statement next. Message IFL505E is sent to the operator's console, from which an eventual response is required. A reply of 'S' to this message stops the transfer of trace blocks to the host module.

Each of the trace blocks received by the host module is time-stamped before being written to the work data set. The time stamp is of the form hh:mm:ss (hour:minute:second) and indicates the time that the trace block was received by the host module. Periodically (for the first trace block, for the last trace block, and for every 200th trace block between) the operator is informed of these time stamps via message IFL508I. A typical IFL508I message might be:

```
IFL508I TRACE BLOCK 15,000 WRITTEN AT 13:40:42
```

This message indicates that the 15,000th trace block was written to the work data set at 1:40:42 p.m. This information may be used when you prepare to print the work data set.

Note: A total of 200 trace blocks is equivalent to approximately 72 pages of printed output (assuming 55 lines per page).

To stop the trace activity the operator must first respond with an 'S' to the message IFL505I (see Appendix E) issued when the trace was initiated. The 'S' response stops the transfer of trace blocks to the host module as soon as the next trace block is received. To stop the trace activity in the controller (which was initiated by the OPTION control statement), a second OPTION statement must be entered. Alternatively, this statement may be entered at the controller's panel.

Note: The trace should not be stopped at the controller's panel until the 'S' response to message IFL505E has been given and accepted.

With the trace activity completed, a readable output listing of the trace blocks can be obtained by entering the PRINT command. This command causes the entire work data set to be formatted and printed. Suppose, however, you are interested only in printing the last portion of the trace blocks. For example, the trace is run to detect a sporadic line error. The trace is stopped when the line error occurs and a printout of only the last portion of the trace blocks is required. To obtain this printout, a PRINT command like the following can be entered:

```
PRINT START=13:40:00
```

This command results in a printout of only those trace blocks written to the work data set after 1:40 p.m.

Utility Control Statements

The dynamic dump utility control statements:

- Obtain a full storage dump
- Dump the trace table area
- Dump trace entries dynamically
- Specify trace options
- Request printing of the information dumped.

In the explanation of each utility control statement, small letters represent parameters for which you supply a value. A combination of capital and small letters, such as in the control statement PRint, represents a statement that may be specified by coding either the entire statement, PRINT, or the truncated portion represented by the capital letters PR.

The DYNADMP Statement

The DYNADMP statement requests a dump of the entire controller storage or a specified portion. The controller does not become idle, and the emulation program need not be reloaded.

Name	*Operation*	*Operands*
`[symbol]`	`DYnadmp`	{`Dynamic` / `Storage` / `Table`}

`[symbol]`

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

`Dynamic`

Specifies that the trace table is to be dumped dynamically as entries are made. This type of dump requires operator intervention to stop the trace. A trace must be started on a communication line via the control panel of the controller or via the dynamic dump facility (the OPTION control statement) before a dynamic trace can be started.

`Storage`

Specifies that the entire contents of controller storage are to be dumped. The emulation program continues processing both during the operation and after the contents have been dumped.

`Table`

Specifies that only the trace table portion of controller storage is to be dumped.

If no operand is specified, a full storage dump is produced.

The DISPLAY Statement

The DISPLAY statement is used to request a display of a portion of controller storage on the operator's console at the host processor.

Name	*Operation*	*Operands*
[symbol]	DIsplay	hhhhh[,n]

[symbol]

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

hhhhh

Specifies the address, in hexadecimal, of the storage to be displayed.

[,n]

Specifies the number of lines (16 bytes of storage per line) to be displayed. The maximum number of lines that can be specified is nine. If *n* is omitted, 1 is assumed.

The PRINT Statement

The PRINT statement requests that a printout (32 bytes of storage per line) of the work data set be sent to the SYSPRINT device (the output data set).

Name	*Operation*	*Operands*
[symbol]	PRint	START=hh:mm:ss

[symbol]

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

START=hh:mm:ss

Specifies that only those trace blocks that were written to the work data set after time *hh:mm:ss* (hour:minute:second) are to be printed.

hh:mm:ss should specify a time that is both later than (or equal to) the time stamp associated with the first trace block and earlier than (or equal to) the time stamp associated with the last trace block that was written to the work data set; otherwise, message IFL510I will be issued to indicate that no trace blocks were found that satisfied the PRINT command.

Note: The print facility correctly interprets a post-midnight time stamp (for a last-written trace block) as later than a pre-midnight time stamp (for a first-written trace block), even though the numeric value of *hh:mm:ss* is lower for the post-midnight time (as, for example, values of 23:55:00 and 00:02:00, representing the seven-minute interval from 11:55 p.m. to 12:02 a.m.)

Example 1: Assume that the first trace block is recorded at 09:05:00 (9:05 a.m.) and the last is recorded at 09:20:00 (9:20 a.m.). The statement PRINT START=09:17:30 would cause printing of trace blocks recorded between 9:17:30 a.m. and 9:20 a.m.

Example 2: Assume that the first trace block is recorded at 23:25:23 (11:25 p.m.), and the last is recorded at 00:40:57 (12:40:57 a.m. the following day). The statement PRINT START=23:40:00 would cause printing of trace blocks recorded between 11:40 p.m. and 12:40:57 a.m.

Example 3: Assume that the first trace block is recorded at 23:25:23 (11:25 p.m.), and the last is recorded at 00:40:57 (12:40:57 a.m. the following day). The statement PRINT START=23:40:00 would cause printing of trace blocks recorded between 11:40 p.m. and 12:40:57 a.m.

If you omit the START operand, the entire work data set will be printed.

Note: If you specify the START operand and there are storage dumps in the work data set with the trace blocks, then these storage dumps are also printed regardless of whether or not they satisfy the START constraint. Storage dumps are not time stamped.

The OPTION Statement

The OPTION statement starts, stops, or alters the program interrupt levels being traced. Level 2 interrupts (line data), or level 3 interrupts (time-out complete or channel data, such as initial selection, data, and status), or both can be traced. Level 1 error log entries are traced continuously after the first level 3 trace is started.

Name	*Operation*	*Operands*
[symbol]	OPtion	F[A]BCDE

`[symbol]`

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

`F[A]BCDE`

Specifies the trace functions desired: start trace or stop trace, program level to be traced, and subchannel to be traced. The values F, A, B, C, D, and E and their meanings are as follows:

F *Function*	*BC* *Data*	*DE* *Bytes*	*Meaning* (L=level)
4	10	xx	Start level 2 trace on subchannel xx.
4	11	xx	Stop level 2 trace on subchannel xx.
4	20	xx	Start level 3 trace on subchannel xx.
4	21	xx	Stop level 3 trace on subchannel xx.
4	30	xx	Start level 2 and level 3 trace on subchannel xx.
4	31	xx	Stop level 2 and level 3 trace on subchannel xx.
4	70	00	Start level 3 trace on trace-defined subchannels.
4	71	00	Stop level 3 trace on trace-defined subchannels.
4	70	FF	Start level 3 trace on all subchannels.
4	71	FF	Stop level 3 trace on all subchannels.

'A' is used when two type 4 channel adapters are installed. A=1 (or A omitted) specifies that the subchannel specified in DE is associated with the first type 4 channel adapter; A=2 specifies that the subchannel specified in DE is associated with the second type 4 adapter.

The PAUSE Statement

The PAUSE statement allows control statements to be entered at the host processor console, after the PAUSE statement is read from the input job stream or entered from the console.

Name	*Operation*	*Operands*
[symbol]	PAuse	none

`[symbol]`

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

The END Statement

The END statement specifies the end of the job and causes termination of the utility after the trace output has been printed.

Name	*Operation*	*Operands*
[symbol]	ENd	none

`[symbol]`

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

The SYSIN Statement

The SYSIN statement is used by the operator to cause control statements to be read from the input stream.

Name	*Operation*	*Operands*
none	SYsin	none

The PARM Field Option

The only PARM field option recognized by the dynamic dump utility is LINECOUNT=nn.

```
PARM='LINECOUNT=nn'   or   PARM='LC=nn'
```

nn

Specifies a number from 10 to 99 which represents the number of lines per page to be printed by the dynamic dump utility in producing its printed output.

If the LINECOUNT parameter is omitted, or if it is given but is syntactically incorrect, then a default of 55 lines per page is assumed.

Example: To specify that the dynamic dump utility print 40 lines on each page of printed output, code the PARM field option as follows:

```
//STEP1 EXEC PGM=IFLSVEP,PARM='LINECOUNT=40'
```

Job Control Statements

The OS job control statements:

- Execute or invoke the program
- Define the output data set, the control statement data set, the work data set, and the communications controller.

The dynamic dump utility operates in a minimum MFT (48K) or MVT (50K) region. It requires no more than four 2314 tracks and one directory block on SYS1.LINKLIB. Work data set requirements depend on the size of the controller being dumped. See Appendix C to determine storage requirements for the work data set.

If a trace entry dump is requested, the work data set must be large enough to hold all of the trace data being dumped. If, however, the work data set is exhausted, the job will abend. It is preferable to use a tape unit for this activity.

The OS job control statements needed to invoke the dynamic dump utility are as follows:

```
//jobname       JOB       (initiates the job.)
//              EXEC      (specifies PGM=IFLSVEP or the procedure name
                          if the job control statements reside in a
                          procedure library.)
//SYSPRINT      DD        (defines a sequential output data set. This
                          data set may be written onto a system output
                          device, a magnetic tape volume, or a direct
                          access volume. The data control block's (DCB)
                          blocksize may be specified [optional].)
//SYSUT1        DD        (defines the communications controller native or
                          IPL subchannel address being used [optional].)
//SYSUT2        DD        (defines a temporary work data set. The
                          contents of the communications controller are
                          written to this data set. DISP=OLD must be
                          specified.)
//SYSIN         DD        (defines the control data set. The data control
                          block's (DCB) blocksize may be specified).
Control statements
/*
//
```

Example of Job Control and Dynamic Dump Utility Statements

The following example shows the statements required to dynamically dump, to the work data set, the entire contents of the controller whose native subchannel address is 007. After the dump is complete, the contents of the work data set are transferred to the output data set and printed. The job ends without operator intervention.

```
//SVEP          JOB       MSGLEVEL=(1,1)
//              EXEC      PGM=IFLSVEP
//SYSPRINT      DD        SYSOUT=A
//SYSUT1        DD        UNIT=007
//SYSUT2        DD        UNIT=2400,VOL=SER=SVTAPE,LABEL=(,NL),
//                        DISP=OLD,DSN=WORK
//SYSIN         DD        *
DUMP            DYNADMP   STORAGE
                PRINT
                END
/*
//
```

Example of Dynamically Dumping Trace Table Entries

The following example shows the statements required to dynamically dump the trace table entries as they are being made in the trace table.

```
//DYNADUMP     JOB        MSGLEVEL=(1,1)
//STEP1        EXEC       PGM=IFLSVEP
//SYSPRINT     DD         SYSOUT=A
//SYSUT1       DD         UNIT=007
//SYSUT2       DD         UNIT=2400,VOL=SER=SVTAPE,LABEL=(,NL),
//                        DISP=OLD,DSN=WORK
//SYSIN        DD *
    PAUSE                 (directs the dynamic dump utility
                          to read control statements from the
                          operator's console.)
/*
//
```

The following is an example of what would appear at the operator's console:

```
                          .
                          .
                          .
IEF403I DYNADUMP STARTED TIME=08.12.16 P00
*IEF233A M 282,SVTAPE,,DYNADUMP,STEP1  P00
@08 IFL501A - REPLY WITH DESIRED FUNCTION OR 'END' P00
!
r 08, 'option 43023'      (this activates level 2 and level 3
                          trace activity in the emulation program
                          on subchannel 23.)
 +IFL503I  FUNCTION COMPLETED - 00  P00
@09 IFL501A - REPLY WITH DESIRED FUNCTION OR 'END' P00
!
r 09,'dy dynamic'         (this starts the transmission of trace
                          entries [64 at a time] to the host module
                          and places them on the work data set)
@10 IFL505E - REPLY 'S' TO STOP TRACE P00
 +IFL508I TRACE BLOCK     1 WRITTEN AT  08:15:24 P00
 +IFL508I TRACE BLOCK   200 WRITTEN AT  08:22:43 P00
 +IFL508I TRACE BLOCK   400 WRITTEN AT  08:31:09 P00
 +IFL508I TRACE BLOCK   600 WRITTEN AT  08:37:58 P00
!
r 10,'s'                  (this stops the transfer of trace blocks
                          to the host module)
 +IFL506I  STOP COMMAND ACKNOWLEDGED P00
 +IFL508I  TRACE BLOCK       712 WRITTEN AT  08:40:12  P00
 +IFL503I  FUNCTION COMPLETED - 00 P00
@11  IFL501A - REPLY WITH DESIRED FUNCTION OR 'END' P00
!
r 11,'print start=08:35:00'(this causes a printout of only
                           those trace blocks written to the
                           work data set after 8:35 A.M.
                           This should be approximately
                           150 blocks)
 +IFL503I  FUNCTION COMPLETED - 00 P00
@12 IFL501A - REPLY WITH DESIRED FUNCTION OR 'END' P00
!
r 12,'option 43123'       (halts trace activity on subchannel 23)
 +IFL503I  FUNCTION COMPLETED - 00 P00
@13 IFL501A - REPLY WITH DESIRED FUNCTION OR 'END'
!
r 13,'end'                (this terminates the DYNADUMP job)
 IEF404I DYNADUMP ENDED  TIME=08.46.07 P00
```

Chapter 6: Generating an Emulation Program under DOS

The emulation program generation procedure under the control of the Disk Operating System (DOS) is a two-stage process.

Stage one of the generation procedure is an assembly job using the communications controller assembler (IFTASM) to prepare, from the program generation macros, an object module (comprising the EP control tables) and a printed assembly listing. You must direct the object module to a sequential file (cards, tape, or direct-access device). Figure 6-1 illustrates the stage one input job stream.

Stage one does not automatically produce the stage two job stream. You must prepare an input job stream that contains the job control and linkage editor statements required as input to stage two. The linkage editor statements comprise the INCLUDE and ENTRY statements appearing at the end of the stage one assembly listing; the sequence in the job stream must match the sequence in the listing.

```
// JOB          jobname
// PAUSE        (before executing stage one,
                assign appropriate source statement
                and private relocatable libraries)
// EXEC         IFTASM
      .
      .
      .
   BUILD        source
      .         program
      .         generation
      .         macro
   GENEND       statements
   END
/&
```

Figure 6-1. DOS Generation Stage One Input Job Stream

Operator intervention is required between the stages of program generation. Diagnostic messages (listed in Appendix D) are printed in the stage one output listing; these indicate any errors that have occurred. If these are serious errors, no object deck is produced. The source statements must be corrected and stage one must be re-executed. If no serious errors occur in stage one, the operator creates stage two.

Stage two of the generation procedure has three steps (see Figure 6-2). In step one, the MAINT utility places the object module produced by stage one on a private relocatable library that you specify. Step two link edits the object module with the preassembled EP object modules specified by the INCLUDE statements mentioned above, and places the resultant load module (phase) on the private core image library that you specify. Step three uses the CSERV utility to move the load module from the core image library to a private sequential load file from which the loader may obtain it.

This input job stream must be prepared by the user and submitted to stage two of the generation procedure.

```
(step  one)
// JOB          jobname
// PAUSE        (Assign appropriate private relocatable
                and core image libraries)

// EXEC         MAINT
    (object deck)
/*
(step  two)
// OPTION       CATAL
   ACTION       MAP,NOAUTO,...
   PHASE        phasename,+0
   INCLUDE      module-1                (these Include and Entry statements
   INCLUDE      module-2                must appear in the same sequence as
     .                                  listed in stage one assembly
     .                                  listing)
     .
   INCLUDE      module-n
   ENTRY        CYASTART
// EXEC         LNKEDT                  (places EP load module [phase] on
                                        private core image library)
(step  three)
   DLBL         IJSYSPH,'phasename'     (defines disk area for load module)
   EXTENT       SYSPCH,...
   ASSGN        SYSPCH,X'xxx'
// EXEC         CSERV                   (moves load module from private core
   PUNCH        phasename               image library to load file specified
/*                                      in preceding DLBL and EXTENT statements)
/&
   CLOSE        SYSPCH,X'xxx'
/&
```

Figure 6-2. DOS Generation Stage Two Input Job Stream

Unresolved External References

If unresolved external references occur in the listing and meet the conditions listed below, they should be ignored.

1. If the trace option is not specified, the following external reference remains unresolved:

   CYATRCEI

2. If the trace option is not specified and the dynamic dump option is specified, the following external references remain unresolved:

   CYASETRC
   CYATABLE

3. For versions of the emulation program where the panel test option is available but not specified, the following external references remain unresolved:

   CYAPANLT
   CYATST10 (Type 1 scanner without auto call)
   CYATST11 (Type 1 scanner with auto call)
   CYATST20 (Type 2 scanner without auto call)
   CYATST21 (Type 2 scanner with auto call)

4. If start-stop terminals are specified but no display terminals are specified, the following external references remain unresolved:

   CYAB28CL
   CYAB2848
   CYAATDA4

5. If start-stop terminals are specified but the DELAY option is not specified, the following external reference remains unresolved:

   CYADAT1

6. If no start-stop terminals are included, the following external references remain unresolved:

Type 2 Scanner	*Type 1 Scanner*
CYAQUIET	CYAQUIET
CYAAATB1	CYAAATB1
CYASTPER	CYASTPER
CYATRN	CYATRN
CYAMTBFR	CYAMTBFR
CYACWRIS	CYACWRIS
CYACREAS	CYACREAS
CYACPOLS	CYACPOLS
CYACPRES	CYACPRES
CYACBRES	CYACBRES
CYACSEAS	CYACSEAS
CYACBKPL	CYACBKPL
CYABTDA0	CYABTDA0
CYACRDCL	CYACRDCL
CYABARP1	CYABARP1
	CYASRCVT
	CYASPCF8
	CYAXSSTT
	CYASPCFA
	CYASPCFB

7. If no binary synchronous terminals are included, the following external references remain unresolved:

Type 2 Scanner	*Type 1 Scanner*
CYARARS0	CYARARS0
CYATAPD0	CYATAPD0
CYATBSWR	CYATBSWR
CYATSTMW	CYATSTMW
CYACWRIB	CYACWRIB
CYACREAB	CYACREAB
CYACADPB	CYACADPB
CYACPREB	CYACPREB
CYACSETB	CYACSETB
CYACPOLB	CYACPOLB
CYACSEAB	CYACSEAB
CYATBSSM	CYATBSSM
CYATBSPR	CYATBSPR
CYATBSPL	CYATBSPL
CYABSHIO	CYABSHIO
CYABSTOP	CYABSTOP
CYATBSRD	CYATBSRD
	CYAPCF45
	CYARCDTA
	CYABPCF8
	CYABPCFA

8. If the dynamic dump option is not specified, the following external reference remains unresolved:

   CYADSTRT

9. If the DELAY operand is not specified in the GROUP macro, the following external reference remains unresolved:

   CYADAT1

10. If no TTY or TWX terminals are included, the following external reference remains unresolved:

    CYATDONE

11. External reference TM598 is defined in the CYASL210/310/220/320 modules.
12. External reference CYAATDA5 is defined in the CYADSP10/20 modules.
13. External reference CYATRCHG has been deleted in the current release of the emulation program.

Chapter 7: Emulation Program Utilities under DOS

This chapter explains the use of the loader, dump, and the dynamic dump utilities under the control of the Disk Operating System (DOS). In order to communicate with the communications controller, the loader, dump, and dynamic dump utility programs require the existence of a native subchannel that is defined by a physical unit block (PUB). A PUB must be created at either system generation time or at initial program load (IPL) time, and must be specified with the appropriate address. See "DOS Requirements for Installing the Dynamic Dump Utility" in the dynamic dump utility section of this chapter for additional DOS considerations. A 3704 or 3705 physical unit block can be generated into any system that includes release 24 (or later release) of DOS.

The Loader Utility

The loader utility program transfers an emulation program load module from the the host processor to the communications controller. It must be run as a job or job step under DOS.

The loader has two modules. One is an operating system utility that may be invoked as any other DOS utility. The other module runs in the communications controller. When the loader is invoked, the controller module is contained within a data area in the host processor loader module. The host processor module loads the controller module into the controller via an initial program load (IPL) command.

The communications controller module of the loader can be executed in any communications controller. The only requirements for the load operation are that the communications controller be identified to the operating system, that it be free to be allocated to the loader job step, and that its power be on. See Appendix C for host storage requirements.

Before the loader utility loads the emulation program into the controller, it loads a diagnostic routine, called the *initial test* routine. This routine tests the communications controller for hardware malfunctions that might later cause failure of the emulation program and sets good parity in EP storage areas. If the initial test routine detects no malfunctions, the loader then loads the emulation program into the controller. If the initial test routine does detect trouble, the routine stops and the loader issues error message IFU004I indicating the fact.

Loading and execution of the initial test routine is optional (it is run unless you specify its omission in the LOAD control statement), but is recommended because it can detect conditions that can later cause failure of the emulation program. Running the initial test routine is especially recommended for a communications controller which has just been powered on because it will set good parity in storage.

Successful completion of the emulation program loading process is indicated to the CPU operator by a write-to-operator message.

Syntax errors in the LOAD statement or permanent I/O errors occurring during loading are indicated by messages sent to the message file (SYSLST).

Messages issued by the loader are given in Appendix F.

Input to the Loader Program

Either two or three work files are used as input to the loader:

- A DASD file (input file) containing the emulation program load module to be loaded.
- A file containing the LOAD control statement specifying the name of the emulation program load module and the communications controller into which the module is to be loaded.
- A file containing the initial test routine (consisting of load modules IFU3705D and IFU3705E) to be loaded before emulation program loading. This file is optional; it may be omitted if the initial test routine is not desired (as indicated by DIAG=NO in the LOAD statement).
- If the initial test routine is desired (as indicated by DIAG=Y6 or Y8) the loader utility requires that interval timer support be present and assigned to the background partition. Prior to loading the communications controller for the first time, two initial test modules must be moved to a direct access file that is accessible by the loader. The CSERV utility program must be used to create this file. The space required for this operation is seven 3330 tracks, ten 3340 tracks, seven 2314 tracks or twenty 2311 tracks.

  The following is an example of a job that accomplishes this task:

```
// JOB    INITTEST
// DLBL   IJSYSPH, other parameters
// EXTENT SYSPCH, other parameters
   ASSGN  SYSPCH,X'xxx', or CLOSE SYSPCH,X'xxx'
// EXEC   CSERV
   PUNCH  IFU3705D,IFU3705E
/*
/&
   CLOSE  SYSPCH, X'xxx'
```

Output from the Loader Program

The loader produces one output file (SYSLST). This file contains the completion messages and/or error messages produced by the loader. See Appendix F for the loader utility messages.

Job Control Statements

The DOS job control statements needed to invoke the loader program are as follows:

`// JOB`		(initiates the job)
`// ASSGN`		(specifies the unit address of the communications controller to be loaded. This statement may be omitted if a permanent assignment exists for the communications controller.)
`// DLBL`		(defines a sequential file that contains a suitably formatted load module.)
`// EXTENT`		
`// ASSGN`		(assigns the data set defined in the previous DLBL and EXTENT statements.)
`// DLBL`	`DIAGFLE,'file-id'`	(defines the sequential file that contains the Initial Test Routine. Required only if DIAG=Y6 or Y8 is specified or implied on any LOAD statement.)
`// EXTENT`	`SYS008,vol.id,1`	
`// ASSGN`	`SYS008,X'ccu'`	
`// EXEC`		(specifies the program name, IFULOAD.)

Utility Control Statement (LOAD)

The DOS program IFULOAD requires one control statement for each communications controller being loaded: the LOAD statement. This statement may not be omitted. It specifies (1) the name of the file that contains the emulation program load module to be loaded, (2) the symbolic address of the communications controller to be loaded, (3) whether or not the initial test routine is to be executed, and (4) the type of direct-access device on which the load file resides.

Name	*Operation*	*Operands*

None	LOAD	`LOADMOD=file name,`
		`3705=SYSxxx`
		`{Y6}`
		`[,DIAG={Y8}]`
		`{NO}`
		`{2311}`
		`[,DEVICE={2314}]`
		`{3330}`
		`{3340}`

`LOADMOD=file name`

Specifies the name of the file that contains the emulation program load module. This file name must be the same as the file name specified in the DLBL statement.

`3705=SYSxxx`

Specifies the symbolic address of the communications controller to be loaded.

DIAG={Y6}
{Y8}
{NO}

Specifies whether the loader is to load the initial test routine into a 3704 or a 3705 without extended addressing (DIAG=Y6), a 3705 with extended addressing (DIAG=Y8), or is not to load the routine at all (DIAG=NO).

DEVICE={2311}
{2314}
{3330}
{3340}

Specifies the type of direct-access device on which the load module resides.

Example of Job and Utility Control Statements

Assume that an emulation program load module residing on a file named LMFR030 is to be loaded into a communications controller with a unit address of 192 and a symbolic address of SYS007. The control and utility statements would be similiar to:

```
// JOB          LOAD
// ASSGN        SYS007,X'007'
// DLBL         DIAGFLE,'IFU3705D,IFU3705E'
// EXTENT       SYS008,PVTLIB,1
// ASSGN        SYS008,X'192'
// DLBL         LMFR030,'EPMOD'
// EXTENT       SYS005,PVTLIB,1
// ASSGN        SYS005,X'192'
// EXEC         IFULOAD
   LOAD         LOADMOD=LMFR030,3705=SYS007,DEVICE=3330,DIAG=Y8
/*
/&
```

In this example, the initial test routine is to be loaded and executed before the emulation program is loaded. The load module resides on a 3330 unit. If the initial test is not wanted, you would specify DIAG=NO in the LOAD statement (or omit the DIAG parameter) and omit the first DLBL, EXTENT, and ASSGN statements.

The Dump Utility

The dump utility program is used to dump the storage contents of a 3704 or 3705 communications controller. A printable copy of the controller's storage contents is produced and placed on a sequential output file (SYSLST).

Executing the dump utility stops operation of the emulation program. Once the dump utility has been started, it should not be canceled before normal completion. If the communications controller contains an active emulation program, the host module writes a message to the operator and waits for a response to verify that the operator wishes to dump the controller. If he does not, he enters a response that ends the dump job.

Refer to Appendix C for the storage requirements of the dump utility.

Dumping the Controller Storage

Dumping from the controller to the direct access work file is performed by the first step of the dump utility. This step first transfers into the communications controller a module containing the utility code needed for the controller to participate in the dumping process. (This module is contained within the dump program in the host processor until transferred to the controller via an initial program load [IPL] command.)

Step one always transfers the entire contents of controller storage and local store registers to the host processor, which places them on a direct access file. However, a small portion of the storage data is overlaid by the dumping process. (The storage area does appear in the listing, but consists of read-only storage [ROS] or dump utility code.) The areas not available are as follows:

For 3705 equipped with a single channel adapter:

Hexadecimal addresses:	Overlaid by:
0 through 1FF	ROS
400 through 4E7	dump utility code
700 through 707	ROS
780 through 79F	ROS

For 3705 equipped with two channel adapters or a 3704:

Hexadecimal addresses:	Overlaid by:
0 through 3FF	ROS
400 through 4E7	dump utility code
700 through 70F	ROS
780 through 79F	ROS

Note: The contents of the controller's external registers are not transferred to the host processor. If the contents of these registers must be examined, they must be displayed on the controller's operator panel and the contents noted before the dump utility is invoked.

When the dumping process is complete, the program informs the CPU operator. At this point the controller is idle and can be reloaded with an emulation program via the loader utility.

For the job control statements needed to both dump and print the contents of controller storage, see the topic below, "How to Dump and Print Storage Contents."

Printing the Dump Data

The second step of the dump utility converts all or a selected part of the dumped data to printable form, then places the data on a sequential output file. The output listing shows the hexadecimal representation of controller storage and register contents, and gives the character equivalents of all EBCDIC bit patterns that represent characters. Beyond this, two options are available, as specified by the DUMP control statement:

- Formatted or unformatted dump of controller storage. The unformatted storage dump is printed with EBCDIC equivalents on the right side of the page. In a formatted storage dump, the mnemonic operation codes are printed with the instructions.
- The complete contents of storage may be listed, or any specified portion or portions of storage.

Utility Control Statement (DUMP)

The dump utility requires one control statement, DUMP. This statement specifies that the dump listing show either the entire contents of communications controller storage or a portion thereof.

Name	*Operation*	*Operands*
[symbol]	DUMP	[FROMADDR=address] [,TOADDR=address] [,MNEMONIC={Y}] {N}

[symbol]

Specifies a name, one to eight characters in length, beginning with an alphabetic character.

FROMADDR=address

Specifies the lower limit of the controller storage to appear on the listing. If you omit FROMADDR, the listing will start at address X'200'. (If you specify a value less than X'200', error message IFU101I is issued and a dump of the entire storage contents is produced.)

TOADDR=address

Specifies the upper limit of communications controller storage to appear on the dump listing. If you omit TOADDR, the listing ends at the upper limit of communications controller storage. (If you specify a value higher than the upper limit of storage, message IFUI01I is issued and a dump of the entire storage is produced.)

MNEMONIC={Y}
{N}

Specifies whether or not the IFLDUMP program is to print the mnemonic operation codes in the dump listing. These codes are printed only if you specify MNEMONIC=Y. When MNEMONIC=Y is specified, the paper output is approximately three times greater than when MNEMONIC=N is specified.

Note: If the controller storage to be dumped contains line trace information, omission of mnemonic operation codes is recommended.

How to Dump and Print Storage Contents

The following is an example of the statements required to dump and print the contents of communications controller storage.

```
// JOB                                  (initiates the job)
// ASSGN     SYS007,X'xxx'              (specifies the unit address
                                        of the communications controller to be
                                        dumped. You may omit this statement if a per-
                                        manent assignment was made for the controller
                                        during EP generation)
// EXEC      IFUDUMP                    (specifies the job step)
```

Note: The logical unit assigned to the communications controller must be SYS007. Use of any logical unit other than SYS007 causes the job to be canceled.

Assume, for example, that a communications controller whose unit address is 019 is to be dumped, that the dump listing is to show the contents of communications controller storage from the beginning through location X'3FFF', and that mnemonic operation codes are to be printed. The control and utility statements would be similar to:

```
//     JOB       DUMP
//     ASSGN     SYS007,X'019'
//     EXEC      IFUDUMP
       DUMP      TOADDR=3FFF,MNEMONIC=Y
/*
/&
```

The Dynamic Dump Utility

The dynamic dump utility is an optional utility that provides the following services that are useful in debugging. This utility can be used to:

- Obtain, without terminating the execution of the emulation program, (1) a storage dump (from location 0 through the end of storage) of the communications controller, or (2) a display, on the operator's console at the host processor, of portions of controller storage (up to 144 bytes) starting at any location, or (3) a dump of the emulation program trace table only.
- Activate or deactivate the emulation program line trace function.
- Obtain a dynamic dump of emulation program trace table entries as they are entered into the trace table.

Input to the Dynamic Dump Utility

Control statements are used to request the various functions of the dynamic dump utility. These control statements may reside in the input job stream or they may be entered via the operator's console.

Initially, the dynamic dump utility reads control statements from the SYSIN file until either an END statement or a PAUSE statement is read. The PAUSE statement instructs the dynamic dump utility to read control statements only from the operator's console until either an END statement or a SYSIN statement is read. The SYSIN statement is the opposite of the PAUSE statement; it instructs the dynamic dump utility to return to the SYSIN file for control statements (beginning with the next statement after the last PAUSE statement). An END statement either encountered in the SYSIN file or entered from the operator's console causes the dynamic dump utility to be terminated.

Output from the Dynamic Dump Utility

Work File—This is a temporary file on which the contents of storage are written. (This file usually resides on a tape unit.)

Output File—This is the file on which the trace or storage dump is printed from the work file. It also contains the dynamic dump control statements and applicable error messages.

Operator's Console—The operator's console at the host processor may receive output as a result of a DISPLAY statement, control statement responses, and error conditions.

Dynamic Dump Operational Characteristics

The dynamic dump utility is used when trouble or error conditions indicate that a dynamic dump of controller storage is desirable to isolate and fix a problem.

The dynamic dump utility physically consists of two modules. One module resides in the host processor (as load module IFUSVEP), and the other module resides in the controller as part of the emulation program. (This module is included in the emulation program only if DYNADMP=YES is specified in the BUILD macro during emulation program generation.) These two modules communicate with each other to transfer specified controller storage to the host module. If the DISPLAY command is used to enter a request, the transferred storage is displayed at the operator's console or sent to the output file; otherwise, the host module will write the received storage to the work file in 516-byte blocks. You may then

invoke the PRINT facility of the dynamic dump utility to print the contents of this work file.

When a particular user request has been satisfied, the host module of the dynamic dump utility issues message IFL503I to inform the operator that the transfer of data to the work file is complete.

Obtaining a Dynamic Dump of Trace Table Entries

The most important function of the dynamic dump utility is its ability to dynamically dump emulation program trace entries. The following paragraphs describe this operation. Refer to the "Utility Control Statements" section for additional information on the control statements discussed and to the "Example of Dynamically Dumping Trace Table Entries," later in this section, for an illustration of the input stream.

To dynamically dump the emulation program trace table entries, first start the trace on the desired range of emulation program subchannels by using the OPTION control statement or by using the communications controller control panel. To initiate the transfer of 516-byte blocks of trace entries (hereafter referred to as trace blocks) to the dynamic dump utility host module, enter the DY Dynamic control statement next. Message IFU505E is sent to the operator's console to inform the operator as to how he can stop the trace activity.

Each of the trace blocks received by the host module is time-stamped before being written to the work file. The time stamp is of the form hh:mm:ss (hour:minute:second) and indicates the time that the trace block was received by the host module. Periodically (for the first trace block, for the last trace block, and for every 200th trace block between), the operator is informed of these time stamps via message IFU508I. A typical IFU508I message might be:

```
IFU508I  TRACE BLOCK  15,000  WRITTEN AT   13:30:42
```

This message indicates that the 15,000th trace block was written to the work file at 1:40:42 p.m. This information may be used when you prepare to print the work file.

Note: A total of 200 trace blocks is equivalent to approximately 72 pages of printed output (assuming 55 lines per page).

To stop the trace activity the operator must first press the console interrupt button. He then must reply in response to the attention routine:

'MSG BG'	(if the dynamic dump utility program is running in the background partition)
or	
'MSG Fx'	(if the dynamic dump utility program is running in the foreground partition 'x')

This action will halt the transfer of trace blocks to the work file as soon as the next trace block is received from the controller and placed on the work file.

To stop the trace activity in the controller (which was initiated by the OPTION control statement), a second OPTION statement must be entered. Alternatively, this may be done at the controller's panel.

With the trace activity completed a readable output listing of the trace blocks can be obtained by entering the PRINT command. This command causes the entire

work file to be formatted and printed. Suppose, however, you are interested only in printing the last portion of the trace blocks. For example, the trace is run to detect a sporadic line error. The trace is stopped when the line error occurs and a printout of the last portion of the trace blocks is required. To obtain this printout, a PRINT command like the following can be entered:

```
PRINT START=13:40:00
```

This will result in a printout of only those trace blocks written to the work file after 1:40 p.m.

Utility Control Statements

The dynamic dump utility control statements:

- Obtain a full storage dump
- Dump the trace table area
- Dump trace entries dynamically
- Specify trace options
- Request printing of the information dumped.

In the explanation of each utility control statement, small letters represent parameters for which you supply a value. A combination of capital and small letters, such as in the control statement PRint, represents a statement that may be specified by coding either the entire statement, PRINT, or the truncated portion represented by the capital letters PR.

The DYNADMP Statement

The DYNADMP statement requests a dump of the entire controller storage or of a specified portion. The controller does not become idle, and the emulation program need not be reloaded.

Name	*Operation*	*Operands*
[symbol]	DYnadmp	{Dynamic / Storage / Table}

`[symbol]`

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

`Dynamic`

Specifies that the trace table is to be dumped dynamically as entries are made. This type of dump requires operator intervention to stop the trace. A trace must be started on a communication line via the control panel of the controller or via the dynamic dump facility (the OPTION statement) before a dynamic trace can be started.

`Storage`

Specifies that the entire contents of controller storage are to be dumped. The emulation program continues processing both during the operation and after the contents have been dumped.

Table

Specifies that only the trace table portion of controller storage is to be dumped.

If no operand is specified, a full storage dump will be produced.

The DISPLAY Statement

The DISPLAY statement is used to request a display of a portion of controller storage on the operator's console at the host processor.

Name	*Operation*	*Operands*
[symbol]	DIsplay	hhhhh[,n]

[symbol]

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

hhhhh

Specifies the address, in hexadecimal, of the storage to be displayed.

[,n]

Specifies the number of lines (16 bytes of storage per line) to be displayed. The maximum number of lines that can be specified is nine. If *n* is omitted, 1 is assumed.

The PRINT Statement

The PRINT statement requests that a printout (32 bytes of storage per line) of the work file be sent to the SYSLST device (the output file).

Name	*Operation*	*Operands*
[symbol]	PRint	START=hh:mm:ss

[symbol]

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

START=hh:mm:ss

Specifies that only those trace blocks written to the work file after time *hh:mm:ss* (hour:minute:second) are to be printed.

hh:mm:ss should specify a time that is both later than (or equal to) the time stamp associated with the first trace block and earlier than (or equal to) the time stamp associated with the last trace block written to the work file; otherwise, message IFU510I will be issued to indicate that no trace blocks were found which satisfied the PRINT command.

Note: The print facility correctly interprets a post-midnight time stamp (for a last-written trace block) as later than a pre-midnight time stamp (for a first-written trace block), even though the numeric value of *hh:mm:ss* is lower for the post-midnight time (as, for example, values of 23:55:00 and 00:02:00, representing the seven-minute interval from 11:55 p.m. to 12:02 a.m.)

Example 1: Assume that the first trace block is recorded at 09:05:00 (9:05 a.m.) and the last is recorded at 09:20:00 (9:20 a.m.). The statement PRINT START=09:17:30 would cause printing of trace blocks recorded between 9:17:30 a.m. and 9:20 a.m.

Example 2: Assume that the first trace block is recorded at 23:25:23 (11:25 p.m.), and the last is recorded at 00:40:57 (12:40:57 a.m. the following day). The statement PRINT START=23:40:00 would cause printing of trace blocks recorded between 11:40 p.m. and 12:40:57 a.m.

Example 3: Assume that the first trace block is recorded at 23:25:23 (11:25 p.m.), and the last is recorded at 00:40:57 (12:40:57 a.m. the following day). The statement PRINT START=00:20:00 would cause printing of trace blocks recorded between 12:20 a.m. and 12:40:47 a.m.

If you omit the START operand, the entire work file will be printed.

Note: If the START operand is specified and there are storage dumps in the work file with the trace blocks, these storage dumps will also be printed regardless of whether or not they satisfy the START constraint. Storage dumps are not time stamped.

The OPTION Statement

The OPTION statement starts, stops, or alters the program interrupt levels being traced. Level 2 interrupts (time-out complete or line data), or level 3 interrupts (channel data, such as initial selection, data, and status), or both can be traced. Level 1 error log entries are traced continuously after the first level 3 trace is started.

Name	*Operation*	*Operands*
[symbol]	OPtion	ABCDE

[symbol]

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

ABCDE

Specifies the trace functions desired: start trace or stop trace, program level to be traced, and subchannel to be traced. The values A, B, C, D, and E and their meanings are as follows:

A Function	*BC Data*	*DE Bytes*	*Meaning*
2	xx	yy	Start level 2 trace on subchannel range xx to yy.
3	xx	yy	Stop level 2 trace on subchannel range xx to yy.
4	10	xx	Start level 2 trace on subchannel xx.
4	11	xx	Stop level 2 trace on subchannel xx.
4	20	xx	Start level 3 trace on subchannel xx.
4	21	xx	Stop level 3 trace on subchannel xx.
4	30	xx	Start level 2 and level 3 trace on subchannel xx.
4	31	xx	Stop level 2 and level 3 trace on subchannel xx.
4	70	00	Start level 3 trace on trace defined subchannels.
4	71	00	Stop level 3 trace on trace defined subchannels.
4	70	FF	Start level 3 trace on all subchannels.
4	71	FF	Stop level 3 trace on all subchannels.

The PAUSE Statement

The PAUSE statement allows control statements to be entered at the host processor console after the PAUSE statement is read from the input stream or entered from the console.

Name	*Operation*	*Operands*
[symbol]	PAuse	none

[symbol]

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

The END Statement

The END statement specifies the end of the job and causes termination of the program after the trace output has been printed.

This statement does not end the trace, however, if trace table entries are being dynamically dumped; in this case the trace must be terminated at the console. Establish operator communication with the host processor for a background partition by pressing the console interrupt button. Reply 'MSG Fx' in reply to the attention routine for foreground partition Fx.

Name	*Operation*	*Operands*
[symbol]	ENd	none

[symbol]

Specifies a name, one to eight characters in length, beginning with an alphabetic character. Do not use a name if you enter statements from the console.

The SYSIN Statement

The SYSIN statement is used by the operator to cause control statements to be read from the input stream.

Name	*Operation*	*Operands*
none	SYsin	none

Job Control Statements

The DOS job control statements:

- Execute or invoke the program
- Define the output file, the control statement file, the work file, and the communications controller.

Work file requirements depend upon the storage size of the controller being dumped. The work file must be a tape unit under DOS. See Appendix C to determine storage requirements for the work file.

The DOS job control statements needed to invoke the dynamic dump utility are as follows:

```
// JOB          [name]     (initiates the job.)
// ASSGN        SYSLST     (defines the output file.)
// ASSGN        SYS011     (defines the communications controller native or IPL
                           subchannel address being used.)
// ASSGN        SYS010     (defines a temporary work file. The contents of the
                           communications controller are written to this file.)
// ASSGN        SYSIPT     (This statement defines the control statement file.)
// EXEC         IFUSVEP    (specifies the job step, IFUSVEP)
   Control statements
/*
/&
```

Example of Job Control and Dynamic Dump Utility Statements

The following example shows the statements required to dynamically dump, to the work file, the entire contents of the controller whose native subchannel address is 001. After the dump is complete, the contents of the work file are transferred to the output file and printed. The job ends without operator intervention.

```
// JOB          SVEP
// ASSGN        SYSLST     (parameters defining the output file)
// ASSGN        SYS010,X'280'
// ASSGN        SYS011,X'001'
// EXEC         IFUSVEP
                DYNADMP STORAGE
                PRINT
                END
/*
/&
```

Example of Dynamically Dumping Trace Table Entries:

The following example shows the statements required to dynamically dump the trace table entries as they are being made in the trace table. The entries go to the work file until operator communication is established. The work file contents are transferred to the output file and then to the printer. The job ends upon completion of the print operation.

```
// JOB          SVEP
// ASSGN        SYSLST          (parameter defining the output file.)
// ASSGN        SYS010,X'280'   (where X'280' represents a device address.)
// ASSGN        SYS011,X'001'   (where X'001' represents the
                                controller address.)
// EXEC         IFUSVEP
   PAUSE                       (Allows operator to enter control
                                statements from console.)
```

The following statements are entered at the host processor console

```
   OPTION       2003F          (Starts trace of level 3 only,
                                starting at address '00' and ending
                                with address '3F'.)
   DYNADMP      DYNAMIC        (Dumps trace table dynamically as
                                entries are made. Establishes
                                operator communication to stop the
                                dynamic dump.)
   SYSIN                       (Returns control to the control
                                statement file.)
```

The remaining statements are read from the control statement file

```
   PRINT                       (PRINT and END, produce a listing
                                on the device specified in the
                                SYSLST statement.)
   END
/*
/&
```

DOS Requirements for Installing the Dynamic Dump Utility

The following Logical Input/Output Control System (LIOCS) modules must be cataloged in the relocatable library:

- IJCFZIW0
- IJDFAZZW
- IJFUZZWZ

The following macros can be assembled to provide the above modules if they are needed:

- CDMOD TYPEFLE=INPUT,WORKA=YES,SEPASMB=YES
- PRMOD CTLCHR=ASA,WORKA=YES,SEPASMB=YES
- MTMOD RECFORM=UNDEF,WORKA=YES,SEPASMB=YES

The controller physical unit block must indicate "TP device" for SVC 27 (HALT I/O) to work. This can be accomplished by specifying 2701 on the ADD command.

The supervisor must be generated with AP=YES for the POST instruction to work, and TP=BTAM or TP=QTAM must be specified for the HALT I/O (SVC 27) instruction to be supported.

If the dynamic dump modules of the host portion of the utility are cataloged in the relocatable library, the following control statements can be used for link-editing them into the core image library.

```
// JOB
// OPTION      CATAL
   PHASE       IFUSVEP,S
   INCLUDE     IFUSVINT
   INCLUDE     IFUSVIND
   INCLUDE     IFUSVCOM
   INCLUDE     IFUSVDMP
   INCLUDE     IFUSVTRF
   ENTRY       IFUSVINT
// EXEC        LNKEDT
//
/&
```

Appendix A: Types of Stations Supported by the IBM 3704 and 3705

The IBM 3704 and 3705 Communications Controllers, when executing an emulation program, can communicate with any of these types of terminals, transmission control units, and computers. The transmission code with which the network control program can communicate with the station is indicated.

Terminals:

IBM 1030 Data Collection System
IBM 1050 Data Communication System[1,2]
IBM 1060 Data Communication System
IBM 2260 Display Station (via IBM 2848 Display Control)
IBM 2265 Display Station (via IBM 2845 Display Control)
IBM 2740 Communications Terminal (Models 1 and 2)[1,2,3]
IBM 2741 Communications Terminal[1,2,3]
IBM 2760 Optical Image Unit (via IBM 2740)
IBM 2770 Data Communications System[4,5]
IBM 2780 Data Transmission Terminal[4,5]
IBM 2972 General Banking Terminal System[4,5]
  IBM 2980 Models 1 and 4 Teller Station
  IBM 2980 Model 2 Administrative Station
IBM 3270 Information Display System[4,5,6]
  IBM 3275 Display Station
  IBM 3277 Display Station
  IBM 3284 Printer
  IBM 3286 Printer
IBM 3650 Retail Store System (in BSC mode)
IBM 3660 Supermarket System (in BSC mode)
IBM 3671 Shared Terminal Control Unit
IBM 3735 Programmable Buffered Terminal[4,5]
IBM 3740 Data Entry System
  IBM 3741 Data Station
  IBM 3747 Data Converter
IBM 3767 Communication Terminal
IBM 3770 Data Communications System
IBM 3780 Data Communication Terminal[4]
IBM 3790 Communications System
IBM 3940 Banking Terminal
IBM 3980 Banking Terminal
IBM Communicating Magnetic Card Selectric® Typewriter

World Trade teleprinters that use CCITT (Consultative Committee on International Telegraphy and Telephony) No. 2 or No. 5 code on leased point-to-point, leased multipoint, or switched network lines.

Terminals using the following line control disciplines: AT & T 83B3 or WU 115A start-stop code, over point-to-point or multipoint leased telegraph lines; WU CPT-TWX (33/35) start-stop code over switched lines.

Attachment of non-IBM terminals is under the provisions of the IBM Multiple Supplier Systems Policy.

Communications Control Units:

IBM 2701 Data Adapter Unit[4,5]
IBM 2703 Transmission Control[4,5]
IBM 2715 Transmission Control Model 2[4]
IBM 3704 Communications Controller[4,5]
IBM 3705 Communications Controller[4,5]

Computers:

IBM System/3[4,5]
IBM System/7[7] (with asynchronous communications adapter) (supported as 2740 Model 1)
IBM System/7[4,5] (with binary synchronous communications adapter) (supported as System/3 [BSC])
IBM System/360 Model 20 (with BSC Adapter)[4,5]
IBM System/360 Model 25 (with Integrated Communications Attachment with Synchronous Data Adapter II)[4,5]
IBM System/370 Model 125 (with Integrated Communications Attachment with Synchronous Data Adapter II)[4,5]
IBM System/370 Model 135 (with Integrated Communications Attachment with Synchronous Data Adapter II)[4,5]
IBM 1130 Computing System (with Synchronous Communications Adapter)[4,5]
IBM 1800 Data Acquisition and Control System (via IBM 1826 Data Adapter Unit with Communication Adapter)[4,5]

---

[1]Binary Coded Decimal (BCD) code
[2]Extended Binary Coded Decimal (Extended BCD) code
[3]Correspondence code
[4]Extended Binary Coded Decimal Interchange Code (EBCDIC)
[5]USA Standard Code for Information Interchange (USASCII)
[6]not supported on switched lines
[7]PTTC/EBCD code

Appendix B: Emulation Program Generation Example

This example is not intended to illustrate the best way to code a configuration, but illustrates the operand formats and shows how they are used.

```
      | BUILD   CA=TYPE1,
                HICHAN=FF,                                  C
                DYNADMP=YES                                 C
                JOBCARD=YES,                                C
                LINETRC=YES,                                C
                LOADLIB=LOD3705,                            C
                LOCHAN=B0,                                  C
                NEWNAME=EP3705P3,                           C
                OBJLIB=OBJ3705,                             C
                QUALIFY=CC,                                 C
                UNIT=SPACE,                                 C
                TYPSYS=OS
        CSB     SPEED=134,                                  C
                WRAPLN=02F,                                 C
                MOD=0,                                      C
                TYPE=TYPE2
        CSB     SPEED=134,                                  C
                WRAPLN=0D0,                                 C
                MOD=1,                                      C
                TYPE=TYPE2
        CSB     SPEED=134,                                  C
                WRAPLN=13F,                                 C
                MOD=2,                                      C
                TYPE=TYPE2
**********      THE FOLLOWING ARE GROUP AND LINE PARAMETERS ********
**********      FOR:  IBM 2260/2848 2400 BPS POINT TO POINT ********
G1      GROUP   CU=2701,                                    C
                CLOCKNG=EXT,                                C
                CUTYPE=2848,                                C
                DUPLEX=FULL,                                C
                FEATURE=(NOIMEND,LRC),                      C
                INTPRI=2,                                   C
                MODEM=OPTION1,                              C
                PAD=NO                                      C
                REPLYTO=2.0,                                C
                TEXTTO=2.0,                                 C
                SPEED=2400
G1L1    LINE    ADDRESS=(020,B0)
G1L2    LINE    ADDRESS=(021,B1)
G1L3    LINE    ADDRESS=(022,B2)
G1L4    LINE    ADDRESS=(023,B3)
G1L5    LINE    ADDRESS=(024,B4)
G1L6    LINE    ADDRESS=(025,B5)
**********      IMS 2260/2848 2400 BPS MULTI-POINT          ********
G2      GROUP   CU=2701,                                    C
                CLOCKNG=EXT,                                C
                CUTYPE=2848,                                C
                DUPLEX=FULL,                                C
                FEATURE=(NOIMEND,LRC),                      C
                INTPRI=2,                                   C
                MODEM=OPTION1,                              C
                PAD=NO                                      C
                REPLYTO=2.0,                                C
                TEXTTO=2.0,                                 C
                SPEED=2400
G2L1    LINE    ADDRESS=(026,B6)
**********      BTAM 2260/2848 4800 BPS POINT TO POINT      ********
G3      GROUP   CU=2701,                                    C
```

```
                CLOCKNG=EXT,                              C
                CUTYPE=2848,                              C
                DUPLEX=FULL,                              C
                FEATURE=(IMEND,LRC),                      C
                INTPRI=3,                                 C
                MODEM=OPTION1,                            C
                PAD=NO                                    C
                REPLYTO=2.0,                              C
                TEXTTO=2.0,                               C
                SPEED=4800
G3L1    LINE    ADDRESS=(027,B7)
G3L2    LINE    ADDRESS=(02B,BB)
G3L3    LINE    ADDRESS=(02C,BC)
**********      BTAM 2260/2848 4800 BPS MULTI-POINT           ********
G4      GROUP   CU=2701,                                  C
                CLOCKNG=EXT,                              C
                CUTYPE=2848,                              C
                DUPLEX=FULL,                              C
                FEATURE=(NOIMEND,LRC),                    C
                INTPRI=3,                                 C
                MODEM=OPTION1,                            C
                PAD=NO                                    C
                REPLYTO=2.0,                              C
                TEXTTO=2.0,                               C
                SPEED=4800
G4L1    LINE    ADDRESS=(02A,BA)
**********      BTAM 2260/2848 2400 BPS MULTI-POINT           ********
G5      GROUP   CU=2701,                                  C
                CLOCKNG=EXT,                              C
                CUTYPE=2848,                              C
                DUPLEX=FULL,                              C
                FEATURE=(NOIMEND,LRC),                    C
                INTPRI=2,                                 C
                MODEM=OPTION1,                            C
                PAD=NO                                    C
                REPLYTO=2.0,                              C
                TEXTTO=2.0,                               C
                SPEED=2400
G5L1    LINE    ADDRESS=(028,B8)
G5L2    LINE    ADDRESS=(029,B9)
G5L3    LINE    ADDRESS=(02D,BD)
G5L4    LINE    ADDRESS=(02E,BE)
**********          BTAM 1050                             ********
G6      GROUP   CU=2702,                                  C
                CLOCKNG=INT,                              C
                DIAL=NO,                                  C
                DUPLEX=FULL,                              C
                FEATURE=(NOIMEND,LRC),                    C
                INTPRI=1,                                 C
                MODEM=OPTION1,                            C
                PAD=NO                                    C
                SPEED=134
G6L1    LINE    ADDRESS=(120,E0),TERM=1050
        LINE    ADDRESS=(121,E1),TERM=1050
    •   LINE    ADDRESS=(122,E2),TERM=1050
        LINE    ADDRESS=(123,E3),TERM=1050
        LINE    ADDRESS=(124,E4),TERM=1050
        LINE    ADDRESS=(125,E5),TERM=1050
    •   LINE    ADDRESS=(126,E6),TERM=1050
        LINE    ADDRESS=(127,E7),TERM=1050
        LINE    ADDRESS=(128,E8),TERM=1050
        LINE    ADDRESS=(129,E9),TERM=1050
    •   LINE    ADDRESS=(12A,EA),TERM=1050
        LINE    ADDRESS=(12B,EB),TERM=1050
        LINE    ADDRESS=(12C,EC),TERM=1050
        LINE    ADDRESS=(12D,ED),TERM=1050
```

```
         LINE    ADDRESS=(12E,EE),TERM=1050
         LINE    ADDRESS=(12F,EF),TERM=1050
      •  LINE    ADDRESS=(130,F0),TERM=1050
         LINE    ADDRESS=(131,F1),TERM=1050
         LINE    ADDRESS=(132,F2),TERM=1050
         LINE    ADDRESS=(133,F3),TERM=1050
         LINE    ADDRESS=(134,F4),TERM=1050
      •  LINE    ADDRESS=(135,F5),TERM=1050
         LINE    ADDRESS=(136,F6),TERM=1050
         LINE    ADDRESS=(137,F7),TERM=1050
         LINE    ADDRESS=(138,F8),TERM=1050
         LINE    ADDRESS=(139,F9),TERM=1050
      •  LINE    ADDRESS=(13A,FA),TERM=1050
         LINE    ADDRESS=(13B,FB),TERM=1050
G6L29    LINE    ADDRESS=(13C,FC),TERM=1050
*********          WRAP LINES                          ********
G7       GROUP   CU=2702,                                C
                 CLOCKNG=INT,                            C
                 DIAL=NO,                                C
                 DUPLEX=FULL,                            C
                 FEATURE=(NOIMEND,LRC),                  C
                 INTPRI=1,                               C
                 MODEM=OPTION1,                          C
                 PAD=NO                                  C
                 SPEED=134
G7L1     LINE    ADDRESS=(02F,FD),TERM=1050
G7L2     LINE    ADDRESS=(0D0,FE),TERM=1050
G7L3     LINE    ADDRESS=(13F,FF),TERM=1050
*****            END EP GEN STAGE 1                          *****
         GENEND  SCANCTL=(0,0,0)
         END
```

Appendix C: Storage Requirements

Utility (Loader, Dump, Dynamic Dump) Storage Requirements

Operating System

Primary storage requirements (OS):

1. With MFT, the utilities operate in a minimum 48K partition.
2. With MVT, the utilities operate in a minimum 50K region.

Auxiliary storage requirements (OS)

1. Residence requirements—SYS1.LINKLIB
   - Loader
     2314 tracks—7 plus 2 directory blocks
     2311 tracks—13 plus 2 directory blocks
     3330 tracks—4 plus 2 directory blocks
     3340 tracks—7 plus 2 direcotry blocks

Note: Includes storage for initial test.

   - Dump and Dynamic Dump
     2314 tracks—3 plus 2 directory blocks
     2311 tracks—6 plus 2 directory blocks
     3330 tracks—2 plus 1 directory block
     3340 tracks—3 plus 1 directory block
2. Work Space Requirements
   - Loader—None
   - Dump and dynamic dump—Depends upon the size controller being dumped. The amount of work data set space may be calculated as follows:

     The number of 512-byte blocks required equals twice the size of the communications controller storage, in K, plus one.

     For example to dump the contents of a controller having 32K bytes of storage required 2(32) + 1 = 65 512-byte blocks (eight 2314 tracks).

Disk Operating System

Primary storage requirements: each utility program operates in 18K.

Auxiliary storage requirements:

1. Core Image Library
   Dump—Six 2311 tracks or three 2314 tracks or two 3330 tracks or three 3340 tracks.
   Loader—Thirteen 2311 tracks or seven 2314 tracks or four 3330 tracks or seven 3340 tracks.
2. Relocatable Library
   Dump—Nine 2311 tracks or five 2314 tracks or three 3330 tracks or five 3340 tracks.
   Loader—eight 2311 tracks or five 2314 tracks or three 3330 tracks or five 3340 tracks.

Note: Initial test not placed in relocatable library.

Emulation Generation Storage Requirements

Operating System

Primary storage requirements (OS):

1. With MFT, the generation procedure operates in a minimum 48K partition.
2. With MVT, the generation procedure operates in a minimum 50K region.

Auxiliary storage requirements (OS)

1. Residence requirements—Requires two-hundred-seventy 2311 tracks or one-hundred-thirty-five 2314 tracks or ninety 3330 tracks or one-hundred-eighteen 3340 tracks in SYS1.MAC3705. Residence requirements for SYS1.OBJ3705 requires fifty 2311 tracks or twenty-five 2314 tracks or seventeen 3330 tracks or twenty-two 3340 tracks.
2. Work space requirements—reflected in assembler storage requirements. See the publication *IBM 3705 Communications Controller Assembler Language,* for additional information on work space requirements.

Disk Operating System

Primary storage requirements—The generation procedure operates in a 20K partition in DOS.

Auxiliary storage requirements (DOS)

1. Residence requirements—Requires fifty-six 2314 tracks or one-hundred-twelve 2311 tracks or thirty-eight 3330 tracks or forty-nine 3340 tracks for the Source Statement Library. Residence requirements for Relocatable Library requires fifty 2311 tracks or twenty-five 2314 tracks or fifteen 3330 tracks or twenty-two 3340 tracks.
2. Work space requirements—Reflected in assembler storage requirements. See the publication *IBM 3705 Communications Controller Assembler Language,* for additional information on work space requirements.

Appendix D: Emulation Program Generation Messages

The communications controller (or OS) assembler program produces diagnostic error messages during expansion of the emulation program generation macro instructions (stage one). The message identifier for each message begins with IFQ.

The format of the messages is as follows:

```
s,IFQnnnI        text
```

s
Is the severity code

A code of 4 is a warning that the condition indicated by the message may cause errors in the emulation program being generated. Generation of the emulation program is not terminated when the severity code is 4.

A code of 8 means that the condition indicated by the message is so severe that the generation procedure cannot continue generating the emulation program. Generation of the stage two job stream is therefore terminated.

IFQ
Is the identifier for emulation program generation messages.

nnn
Is the message serial number

I
Indicates that the message is for information. No action by the operator is required; programmer action may be required.

text
Is the text of the message explaining the error condition.

```
                              {REQUIRED.}
IFQ001I  aaa=bbb  INVALID,    {ccc IS ASSUMED.}
                              {IGNORED.}
```

Explanation:
bbb is not a valid specification for operand *aaa.*

System Action:
One of the following occurs:

- Operand *aaa* is required. Generation is terminated.
- The generation procedure assumes the default value *ccc.*
- Operand *aaa* is not required; value *bbb* is ignored.

User Action:
If operand *aaa* is required or the default value *ccc* is not acceptable, correct operand *aaa* and resubmit stage one.

```
IFQ002I  yyy INVALID NAME, EXCEEDS 8 CHARACTERS, IGNORED.
```

Explanation:
The symbol *yyy* specified in the name field of the macro instruction exceeds 8 characters.

System Action:
The name is not required and is ignored.

User Action:
None.

```
IFQ003I  yyy INVALID NAME, FIRST CHARACTER NOT ALPHABETIC, IGNORED.
```

Explanation:
The symbol *yyy* specified in the name field has a non-alphabetic first character.

System Action:
The name is not required and is ignored.

User Action:
None.

```
                                {REQUIRED.}
IFQ005I  bbb-INVALID SUBOPERAND, {ccc IS ASSUMED.}
                                {IGNORED.}
```

Explanation:
The value *bbb* is an invalid suboperand value.

System Action:
The generation procedure takes one of these actions:

- Suboperand *bbb* is required. Generation is terminated.
- The generation procedure assumes the default value *ccc.*
- The invalid suboperand value is ignored.

User Action:
If suboperand *bbb* is required or the default value *ccc* is not an acceptable value, correct operand *bbb* and resubmit stage one.

IFQ006I SEQUENCE ERROR-mmm,... NOT DEFINED [,explanation]

Explanation:
The macro or macros *mmm* do not appear in the emulation program generation input statements, or they appear in incorrect sequence. The macro or macros specified must precede the macro being processed (that is, the macro for which this message appears). The explanation, if any, describes the conditions that require the macro or macros.

System Action:
Generation is terminated.

User Action:
Insert each macro or macros *mmm*, in the correct sequence in the input statements and resubmit stage one.

IFQ007I SEQUENCE ERROR-mmm1 PRECEDES mmm2.

Explanation:
Macro *mmm2* does not precede macro *mmm1* in the emulation program input statements. Macro *mmm2* must precede macro *mmm1*.

System Action:
Generation is terminated.

User Action:
Correct the sequence of macros *mmm1* and *mmm2* and resubmit stage one.

IFQ008I TERM=bbb, NON-SUPPORTED TERMINAL TYPE.

Explanation:
The type of terminal indicated by *bbb* is not a type supported by the emulation program.

System Action:
Generation is terminated.

User Action:
Correct the value *bbb* and resubmit stage one.

```
                                          {IGNORED.}
IFQ009I aaa=bbb INVALID, NOT WITHIN RANGE, {ccc IS ASSUMED.}
                                          {REQUIRED.}
```

Explanation:
The value *bbb* specified for operand *aaa* is not within the valid range of values.

System Action:
The generation procedure takes one of these actions:

- Operand *aaa* is not required and is ignored. Generation continues.
- The default value *ccc* is assumed. Generation continues.
- A value is required. Generation is terminated.

User Action:
If a value is required or the value assumed is not acceptable, correct the value and resubmit stage one.

```
IFQ011I  bbb PREVIOUSLY SPECIFIED,  {IGNORED.}
                                    {REQUIRED.}
```

Explanation:
Suboperand value *bbb* was specified more than once for the operand being processed.

System Action:
The generation procedure takes one of these actions:

- A value is required. Generation is terminated.
- Suboperand *bbb* is not required and is ignored. Generation continues.

User Action:
If the omission of the duplicate value does not provide the correct value for the operand, specify the correct value or values in the operand and resubmit stage one.

```
IFQ013I  PARAMETERS CONFLICT, explanation.
```

Explanation:
One or more specified parameters conflict. The explanation defines the conflicting parameters.

System Action:
The action taken is described in the explanation portion of the message.

User Action:
If the action taken, as described in the explanation, is not acceptable, stage one should be resubmitted with the parameters respecified such that a conflict no longer exists.

IFQ0014I bbb BPS-NON STANDARD.

Explanation:
The value *bbb* specified as one of the data rates (bits per second) for the CSB macro being processed is not one of the standard data rates for a communications scanner.

System Action:
The data rate *bbb* is accepted as valid. Generation continues.

User Action:
If one of the standard data rates for a communication scanner was intended, resubmit stage one with the correct value. If the value indicated was intended, no action is required.

IFQ027I aaa NOT SPECIFIED, REQUIRED, explanation.

Explanation:
Operand *aaa* is not specified and is required for the reason given in the explanation part of the message.

System Action:
Generation is terminated.

User Action:
Specify a value for operand *aaa* and resubmit stage one.

IFQ028I aaa=bbb INVALID, EXCEEDS n CHARACTERS.

Explanation:
Operand *aaa* is specifed as having the value *bbb*. The number of characters specified in *bbb* exceeds the maximum, *n*, allowed for the operand.

System Action:
If the severity code is 4, *bbb* is ignored and processing continues. If the severity code is 8, generation is terminated.

User Action:
If a value for *aaa* is required or the severity code is 8, resubmit stage one with the correct value for operand *aaa*.

```
                                                          {IGNORED.}
IFQ029I  aaa=bbb INVALID, FIRST CHARACTER NOT ALPHABETIC, {ccc IS ASSUMED.}
                                                          {REQUIRED.}
```

Explanation:
In operand *aaa*, the first character of *bbb* is not alphabetic.

System Action:
The generation procedure takes one of the following actions:

- Operand *aaa* is not required and the value *bbb* is ignored. Generation continues.
- The default value *ccc* is assumed.
- Operand *aaa* is required. Generation is terminated.

User Action:
If operand *aaa* is required, or the default value *ccc* is not acceptable, correct operand *aaa* and resubmit stage one.

```
IFQ030I  aaa NOT SPECIFIED {ccc IS ASSUMED.}
                           {REQUIRED.}
```

Explanation:
Operand or suboperand *aaa*, which is required, is not specified.

System Action:

The generation procedure takes one of the following actions:

- The default value *ccc* is assumed.
- Operand *aaa* is required. Generation is terminated.

User Action:
If the default value *ccc* is unacceptable, specify the correct value for operand (suboperand) *aaa* and resubmit stage one.

```
IFQ032I  mmm HAS PREVIOUSLY BEEN ENTERED-IGNORED.
```

Explanation:
Macro *mmm* is specified more than once in the emulation program generation input statements. This macro may be specified only once.

System Action:
The specification of macro *mmm* currently being processed is ignored. Generation continues.

User Action:

- If the first appearance of macro *mmm* in the input statements is correct, no action is required.
- If the second appearance of macro *mmm* in the input statements is correct, move the macro statement to the correct position in the input statements, and remove the first appearance of the macro. Then resubmit stage one.

```
            {GROUPS}
IFQ033I  NO {LINES}  DEFINED IN THIS GENERATION.
```

Explanation:
The program source statements do not contain any GROUP or LINE macros. GROUP or LINE macros are required.

System Action:
Generation is terminated.

User Action:
Add to the program source statements the GROUP and LINE macros required to define the teleprocessing network.

```
IFQ035I  CSB MOD=bbb SPECIFIED, ALL LOWER CSBs REQUIRED.
```

Explanation:
MOD=*bbb* is specified in the CSB macro, but one or more CSB macros specifying a lower value for MOD is missing.

System Action:
Generation is terminated.

User Action:
Insert in the emulation program generation input statements, preceding the CSB macro in which MOD=*bbb* is specified, one or more CSB macros having lower values for the MOD operand. Resubmit stage one.

```
IFQ036I  SPEED=bbb INVALID, EXCEEDS MAXIMUM SPEED FOR TYPE1 CS, REQUIRED.
```

Explanation:
bbb is specified as the line speed; this speed exceeds the maximum speed (7200 bps) at which a line attached to a type 1 communication scanner can operate.

System Action:
Generation is terminated.

User Action:
Specify the correct speed and resubmit stage one.

```
IFQ038I  SPEED=bbb INVALID, CS OSCILLATOR SPEED LESS THAN ONE-HALF OF LINE
         SPEED NOT FOUND, REQUIRED FOR EXTERNAL CLOCKING.
```

Explanation:
SPEED=*bbb* is specified for this line which has external clocking. No speed less than one-half *bbb* was defined for the communications scanner to which this line is attached.

System Action:
Generation is terminated.

User Action:
Correct one of the following and resubmit stage one:

1. If the line is attached to the wrong scanner, respecify ADDRESS.
2. If the CSB macro specifies the wrong oscillator bit rates, respecify the CSB macro.
3. If *bbb* is incorrect, respecify SPEED.

```
IFQ039I  SPEED=bbb NOT CHECKED FOR OSCILLATOR ASSOCIATION, explanation.
```

Explanation:
bbb is specified as the speed for this line, but it cannot be checked for validity against the communications scanner oscillator rates because of the reason shown in the explanation.

System Action:
Generation is terminated.

User Action:
Correct the errors shown in the explanation and resubmit stage one.

```
IFQ040I  SPEED=bbb INVALID, CSB SPEED EQUAL LINE SPEED NOT FOUND, REQUIRED FOR
         INTERNAL CLOCKING.
```

Explanation:
SPEED=*bbb* is specified for this line which has internal clocking. No speed equal to *bbb* was defined for the scanner to which this line is attached.

System Action:
Generation is terminated.

User Action:
Correct one of the following and resubmit stage one:

1. If the line is attached to the wrong scanner, respecify ADDRESS.
2. If the CSB macro specifies the wrong oscillator rates, respecify the CSB macro.
3. If *bbb* is incorrect, respecify SPEED.

```
IFQ051I  LINE CONFLICT-SPEED GT speed AT address1-address2 AND
         LINE AT address3-address4.
```

Explanation:
During the automatic resolution of the line scan limits, a line with speed greater than *speed* was found in the range of addresses *address1-address2* and another was found in the range of addresses *address3-address4*. This is an invalid configuration, and no valid scan limit can be set.

System Action:
Generation is terminated.

User Action:
Either:

- If the configuration is incorrect, respecify the SPEED and/or ADDRESS and/or AUTO operands, and resubmit stage one or
- Override the automatic scan limit processing by specifying the SCANCTL operand on the GENEND macro and resubmit stage one.

```
IFQ052I  LINE address1, CONFLICTS WITH:  address2,...
```

Explanation:
Line address *address1* requires the use of address substitution. If address substitution is performed, any of the lines *address2,...* that are defined will be disabled.

System Action:
Generation is terminated.

User Action:
Either:

- Respecify the configuration of lines to remove the conflict, and resubmit stage one; or
- Override the automatic substitution processing by specifying the SCANCTL operand on the GENEND macro and resubmit stage one.

IFQ100I xxx NOT REQUIRED FOR DOS, IGNORED.

Explanation:
Parameter *xxx* was specified, but is not used, in a DOS generation.

System Action:
The parameter is ignored.

User Action:
If TYPSYS is correct, no action is required. If TYPSYS is incorrect, specify the correct system and resubmit stage one.

IFQ105I TERM=bbb INVALID, SHOULD BE CODED AS CUTYPE=bbb, REQUIRED.

Explanation:
Terminal type *bbb* is not a supported terminal type but is a valid control unit type (CUTYPE).

System Action:
Generation is terminated.

User Action:
Specify *bbb* as a CUTYPE and resubmit stage one.

IFQ106I WRAPLN=bbb NOT CHECKED FOR MOD ASSOCIATION, explanation

Explanation:
Address *bbb* is specified but it cannot be checked for validity because of the reason shown in the self-defining explanation.

System Action:
Generation is terminated.

User Action:
Correct the error shown in the explanation and resubmit stage one.

IFQ107I SPEED=bbb INVALID, EXCEEDS MAXIMUM OF 2400 FOR INTERNAL CLOCKING, REQUIRED.

Explanation:
bbb is specified as the speed for this line. Internal clocking is specified so the maximum allowed is 2400.

System Action:
Generation is terminated.

User Action:
Specify the correct speed and resubmit stage one.

```
IFQ108I  chanaddr=bb NOT CHECKED FOR LOCHAN-HICHAN ASSOCIATION, ERROR IN LOCHAN
         OR HICHAN IN THE BUILD MACRO
```

Explanation:
bb is not checked against the LOCHAN-HICHAN range of subchannels because of an error in the LOCHAN or HICHAN operand in the BUILD macro.

System Action:
The subchannel address *bb* is accepted as a value.

User Action:
Specify the correct value for LOCHAN or HICHAN and resubmit stage one.

```
IFQ110I  FEATURE=ffff IS STANDARD FOR TERMINAL TYPE bbb.
```

Explanation:
Feature *ffff* is standard when terminal *bbb* is specified in the LINE Macro. The FEATURE=*ffff* suboperand does not have to be specified.

System Action:
None.

User Action:
None.

```
IFQ111I  SEQUENCE ERROR, PREVIOUS GROUP HAS NO LINES.
```

Explanation:
There are no LINE macros following the preceding GROUP macro.

System Action:
Generation is terminated.

User Action:
Correct the sequence of macros so at least one LINE macro follows the preceding GROUP macro, or remove the GROUP macro.

```
        {HICHAN}
IFQ112I {LOCHAN}=aaa IS NOT COMPATIBLE WITH THE PLUGGABLE HARDWARE SUBCHANNELS
```

Explanation:

LOCHAN address *aa* is not a subchannel equal to 16n where $n \geq 0$. When subchannels are installed in the communications controller, the lowest subchannel installed will have a subchannel number equal to 16n.

HICHAN address *aa* is not a subchannel equal to 4n–1 where $n \geq 1$. When subchannels are installed in the communications controller, the highest subchannel installed will have a subchannel number equal to 4n–1.

System Action:

None.

User Action:

Correct the HICHAN or LOCHAN value so that the highest or lowest subchannel specified is consistent with the subchannels installed. A subchannel installed in the controller and used by the host processor, but not specified to the control program in the program source statements produces unpredictable results.

```
IFQ113I  OVER 20 DUALCOM PAIRS HAVE BEEN SPECIFIED
```

Explanation:

Over 20 DUALCOM pairs are specified in the program generation source statements. The generation procedure verifies proper pairing of a maximum of 20 DUALCOM pairs. This is a limit of the verification mechanism only. More DUALCOM pairs are allowed but not completely checked.

System Action:

None.

User Action:

Ensure that this line has a matching DUALCOM line as explained in the DUALCOM operand description under the LINE macro.

Appendix E: OS Utility Messages

This appendix shows the format of each error, warning, and completion message issued to the programmer or to the operator of the host processor CPU during execution of the loader, dump, and dynamic dump utility programs under the control of the operating system.

Messages Issued by the Loader

```
IFL000I  ERROR--LOADING PROCESS TERMINATED ** (ddname) COULD NOT BE OPENED **
```

Explanation:
The data set indicated by *ddname* could not be opened (message IFL010I is also sent). If the DD statement is missing, another system message (IEC130I) will also be issued.

Utility Action:
The loading process is terminated.

User Action:
Ensure that the indicated DD statement has the correct specification and that the 3704= or 3705= parameter specifies the proper *ddname.*

```
IFL001I  UTILITY END   xx WAS THE HIGHEST SEVERITY CODE
```

Explanation:
The loader utility has processed all of the control cards in the input data set. The severity codes (*xx*) possible are:

00 The loading process was completed successfully; all controllers that were to be loaded are now loaded.

04 The loading process for at least one of the controllers to be loaded generated a warning or error message.

08 Because of a severe error, none of the controllers to be loaded was successfully loaded.

Utility Action:
The loader job is completed.

User Action:
If the severity code is greater than zero, examine the message data set for the appropriate messages. Correct the job control statements in error and resubmit the job. Any load procedures that were successfully completed do not need to be resubmitted.

IFL002I ERROR-LOADING PROCESS TERMINATED ** LOADMOD RECORD SIZE TOO LARGE **

Explanation:
The input record size of the emulation program load module was too large for the buffer space available in the host processor (message IFL010I is also sent).

Utility Action:
The controller to be loaded with the indicated load module is not loaded; the loader utility processes the next utility control card, if any.

User Action:
Link-edit the load module again specifying the 'DC' parameter to assure proper load module record size and resubmit the loader job.

IFL003I ERROR--LOADING PROCESS TERMINATED ** SYSUT1 BLDL ERROR **

Explanation:
The build list function (BLDL system macro) failed for the emulation program load module member of the SYSUT1 data set. Either the load module was not found, or a permanent I/O error occurred when the directory was searched. (Message IFL010I is also sent.)

Utility Action:
The controller to be loaded with the indicated load module is not loaded, the loader utility processes the next utility control card, if any.

User Action:
Ensure that the LOADMOD parameter of the load utility control card specifies the proper load module name and that the load module having that name is a member of the SYSUT1 data set.

IFL004I ERROR--LOADING PROCESS TERMINATED ** (ddname) PERMANENT I/O ERROR **

Explanation:
A permanent I/O error occurred in the communications controller during loading (message IFL010I is also sent).

Utility Action:
Loading of the controller is terminated; the loader processes the next utility control card, if any.

User Action:
Resubmit the loader job.

Note: This message is preceded by a SYNAD message.

```
IFL005I  ERROR--LOADING PROCESS TERMINATED
         ** INITIAL-TEST DETECTED 3704/3705 ERROR **
```

Explanation:
The initial test routine did not return control to the loader utility. This indicates that a hardware error occurred on the controller that would prevent the emulation program from executing properly (message IFL010I is also sent).

Utility Action:
The controller is not loaded; the loader utility processes the next utility control card, if any.

User Action:
The operator may follow the problem determination procedure.

```
IFL006I  ERROR--LOADING PROCESS TERMINATED ** CONTROL STATEMENT ERROR **
```

Explanation:
The LOAD utility control card contained a syntax error. (This message is sent to both SYSPRINT and the operator.)

Utility Action:
The controller is not loaded, the loader utility processes the next utility control card, if any.

User Action:
Correct the erroneous LOAD card and resubmit the loader job stage.

```
IFL007I  ERROR--LOADING PROCESS TERMINATED ** PROGRAM FAILURE IN 3704/3705 **
         ** PROGRAM FAILURE IN 3704/3705 ***
```

Explanation:
The controller module of the loader utility encountered a software or hardware error in the controller (message IFL010I is also sent).

Utility Action:
Loading the controller is terminated; the loader utility processes the next utility control card, if any.

User Action:
The operator may follow the problem determination procedure.

IFL008I LOAD COMPLETE 3704/3705=xxx LOADMOD=(member).

Explanation:
The controller whose unit address is *xxx* was successfully loaded with the emulation program load module whose member name is specified by *(member)*

Utility Action:
The loader utility processes the next control card, if any.

User Response:
None.

IFL009I WARNING--LOADING PROCESS COMPLETED ** LOAD MODULE LARGER THAN 3704/3705 **

Explanation:
The load module is too large for the controller. The loader utility loaded as much of the load module as possible in the controller and attempted to give control to that load module. Either the LOADMOD parameter of the LOAD utility control card specified the wrong load module member name for the controller specified by the 3705= parameter, or the emulation program specified is too large for the controller and must be reduced in size.

Utility Action:
The loader utility processes the next utility control card.

User Response:
Correct the LOAD utility control card or regenerate an emulation program of a size the controller can accommodate.

IFL010I LOAD FAILED 3704/3705-xxx LOADMOD=(member).

Explanation:
The loading process for the controller indicated by *xxx* has failed. This message is sent only to the operator, via a Write-to-Operator (WTO) command.

Utility Action:
The loader utility processes the next control card, if any.

User Response:
Examine the SYSPRINT output for messages defining the problem and respond accordingly.

IFL011I ERROR--LOADING PROCESS TERMINATED ** -MISSING KEYWORD- **

Explanation:
A required keyword parameter is missing from the LOAD utility control card. (Message IFL006I is also sent to the operator.)

Utility Action:
The loader utility processes the next LOAD utility control card, if any.

User Action:
Correct the erroneous LOAD card and resubmit the loader job.

IFL012D 3704/3705-xxx ACTIVE ** REPLY TO CONTINUE **

Explanation:
An attempt has been made to load the *xxx* communications controller which contains an active control program.

Utility Action:
The loader utility waits for the operator's reply.

User Action:
If the controller should be loaded, enter `REPLY xx,'U'`. This causes the loader utility to continue the load for this controller. If the controller should not be loaded, enter `REPLY xx,'M'` to terminate the load request; processing continues with the next request.

IFL013I 3704/3705 ACTIVE--LOAD CANCELED BY THE OPERATOR.

Explanation:
The controller was in an active state, and the operator chose not to continue the load.

Utility Action:
The loader utility continues with the next request, if any.

User Action:
None

IFL014I ERROR--LOADING PROCESS TERMINATED
** UNEXPECTED END-OF-FILE ON MEMBER xxxxxxxx **

Explanation:
An end-of-file condition occurred on the indicated member before the load

module and record produced by the linkage editor were found. If the member name is IFL3705x, the problem exists with the initial test routine on SYSUT3; otherwise, the problem exists with the LOADMOD member on SYSUT1.

Utility Action:
The loader continues with the next request.

User Action:
Check the link-edit of the indicated member to ensure successful completion. If link edit was not successful, repeat the link-edit step and request the load operation again.

IFL015I ERROR--LOADING PROCESS TERMINATED ** 3704/3705 DEVICE TYPE CONFLICT **

Explanation:
The specified 3705 device is not identified as a 3704 or 3705 by the operating system.

Utility Action:
The loading process continues with the next control card.

User Action:
Ensure that the DD card describing the 3704 or 3705 has the correct unit and that DIAG= is specified correctly for this particular communications controller.

IFL100I xxxxxx CAN NOT BE OPENED.

Explanation:
The named data set referred to by *xxxxxx* (SYSUT1 or SYSUT2) could not be opened. Either the DD statement that defines the data set is not included in the input stream or a DCB parameter is invalid. If the DD statement is missing, another system message (IEC130I) will also be present.

Utility Action:
The job is terminated.

User Action:
Ensure that a DD statement for SYSUT1 and SYSUT2 is included in the input stream and that the parameters on the DD statement are correct. Resubmit the job.

IFL102I 3704/3705-xxx COULD NOT BE DUMPED.

Explanation:
An error has occurred, and none of the contents of the controller could be copied to the work data set.

Utility Action:
The dump job is terminated.

User Action:
Ensure that the data set characteristics accurately describe the data set being used. If they do not, correct the characteristics and resubmit the job. If the error persists, keep the listings and call IBM.

IFL103I 3704/3705-xxx HAS BEEN DUMPED SUCCESSFULLY.

Explanation:
The contents of the controller whose unit address is *xxx* have been transmitted to the work data set (SYSUT2).

Utility Action:
The dump utility program will next interpret the dump control cards and produce the requested outputs.

User Action:
None.

IFL104I [self explanatory]

Explanation:
Text is produced by a SYNAD macro.

IFL112D 3704/3705-xxx ACTIVE *** REPLY TO CONTINUE ***

Explanation:
An attempt has been made to dump the controller indicated by *xxx* but this controller is currently executing a program.

Utility Action:
The operator's reply is awaited.

User Action:
If dumping the controller is desired (thereby terminating execution of its program), the operator should respond `REPLY xx, 'U'` to this message. Otherwise, he should respond `REPLY xx, 'M'` or cancel the job.

IFL200I XXXXXXXX CAN NOT BE OPENED

Explanation:
The named data set (SYSUT2, SYSPRINT, or SYSIN) could not be opened. Either the DD statement that defines the data set is not in the job stream, or a DCB parameter is invalid.

Utility Action:
The job step is terminated.

User Action:
This is a probable user error. Ensure that a DD statement for SYSUT2, SYSPRINT, and SYSIN is included in the input stream, and that the parameters on the DD statements are correct. Resubmit the job.

IFL201I INVALID CONTROL STATEMENT; DEFAULT TAKEN.

Explanation:
The DUMP statement is incorrectly coded.

Utility Action:
The dump utility provides an unformatted dump of the entire contents of the controller.

User Action:
Probable user error. If the unformatted dump provided does not provide sufficient information, correct the DUMP statement and resubmit the job.

IFL204I THE EMULATOR ERROR LOG IS EMPTY.

Explanation:
There are no error messages in the emulator error log.

IFL205I CONTROL STATEMENT MISSING; DEFAULT TAKEN.

Explanation:
No DUMP control statement was provided.

Utility Action:
An unformatted dump of the entire storage is produced.

User Action:
None.

IFL500I SYSUT1 NOT OPENED.

Explanation:
SYSUT1 could not be opened. The DD statement defining the data set was not included in the input stream.

Utility Action:
The dynamic trace function is terminated.

User Action:
Probable user error. Ensure that a DD statement for SYSUT1 is included in the input stream and that the parameters on the DD statement are correct. Resubmit the job.

IFL501A REPLY WITH DESIRED FUNCTION OR END.

Explanation:
The dynamic dump utility is requesting control statement input from the operator because there was no input stream or because a PAUSE statement was encountered in the input stream.

Utility Action:
The utility waits for the operator's reply.

User Action:
Enter a control statement. If all desired functions are complete, enter `REPLY xx,'END'`. If more control statements are in the input stream, enter `REPLY xx, 'SYSIN'`.

IFL502I THE FUNCTION COULD NOT BE PERFORMED.

Explanation:
A permanent input/output (I/O) error was encountered while processing SYSUT1.

Utility Action:
The utility terminates the job step with a 'USER 0001' abend code.

User Action:
Ensure that DYNADMP=YES is specified in the BUILD macro, and that the unit allocated to SYSUT1 is the type 1 channel adapter of the communications controller.

IFL503I FUNCTION COMPLETED - nn.

Explanation:
The function has completed with the value indicated by *nn.*

Utility Action:
The function is terminated.

User Action:
If *nn* is 00, completion is normal and no action is required. If *nn* is 08, ensure that the control statements appear in the proper sequence. The function was not acceptable to the emulation program.

IFL504I INVALID CONTROL STATEMENT

Explanation:
A control statement was incorrectly specified.

Utility Action:
The function is terminated.

User Action:
Probable user error. Ensure that the control statement is valid and retry the function.

IFL505E REPLY 'S' TO STOP TRACE.

Explanation:
To stop the dynamic dumping of trace table entries from the controller to the work data set, reply 'S' to this message.

Utility Action:
The dynamic dump utility will reissue this message if a response other than 'S' is entered. 'S' is the only response that will stop the transfer of trace blocks to the host module; it will not stop the entire trace activity (the transfer of trace entries to the trace table) in the communications controller. To stop the entire trace, the user should enter an OPTION command with A=3 specified.

User Action:
When it has been determined that enough trace data has been collected, the user should reply 'S' to this message. When this has been acknowledged, the user may want to enter an OPTION command with A=3 specified to terminate the trace activity in the controller.

IFL506I STOP COMMAND ACKNOWLEDGED.

Explanation:
The dynamic dump utility has acknowledged the 'S' response to message IFL505E.

Utility Action:
The transfer of trace blocks to the host module of the dynamic dump utility has been stopped.

User Action:
None

IFL507I FUNCTION NOT AVAILABLE OR INVALID.

Explanation:
The desired function was not available due to unavailable devices or was found invalid by the controller portion of the dynamic dump utility.

Utility Action:
The function is terminated.

User Action:
Probable user error. Ensure that the control statement is valid and that required devices are available, then retry the function.

IFL508I TRACE BLOCK nnn,nnn WRITTEN AT hh:mm:ss.

Explanation:
Trace block number nnn,nnn was written to the work data set at time hh:mm:ss (hour:minute:second).

Utility Action:
All trace blocks are time-stamped before being written to the work data set. For the first trace block, for the last trace block, and for every 200th trace block in-between, the message IFL508I is sent to the host processor console.

User Action:
Refer to the IFL508I messages when printing the work data set. The PRINT facility permits selective printing according to time.

IFL509I INVALID PARM FIELD ON EXEC CARD - DEFAULTS ASSUMED.

Explanation:
An option specified in the PARM field of the EXEC card was misspelled or was syntactically incorrect.

Utility Action:
Default values for the options are assumed and processing continues.

User Action:
Refer to the topic "The Parm Field Option" in the dynamic dump utility section of this manual as an aid to determining the error.

IFL510I NO TRACE BLOCKS SATISFY THE PRINT COMMAND.

Explanation:
No trace blocks were found that were written to the work data set after the time specified (by START=) in the PRINT control statement.

Utility Action:
The dynamic dump utility does not print any trace blocks. If, however, there is data other than trace blocks on the work data set (for example, storage dumps), this nontrace data will be printed.

User Action:
Refer to the IFL508I messages sent to the operator's console during the time the trace was taken. These will indicate the range of timestamps associated with the trace blocks written to the work data set. From this information, derive a meaningful START= time to specify on the PRINT command.

IFL511I EP-DYNADMP IS UNDER THE CONTROL OF ANOTHER HOST

Explanation:
The EP portion of the dynamic dump utility is busy with the dynamic dump requests of another host processor. (This message also occurs if an incorrect subchannel address is specified in the SYSUT1 statement.)

Utility Action:
Message IFL512A is issued.

User Action:
None.

IFL512I REPLY 'YES' TO CONTINUE UNCONDITIONALLY OR 'NO' TO END

Explanation:
This message follows message IFL511I and requires an operator response to indicate the utility action desired. The emulation program can transfer dynamic dump data over only one subchannel at a time. Therefore, a dynamic dump operation over one subchannel must end before a dynamic dump operation over

a second subchannel can begin. The operation over the first subchannel can end normally or it can be aborted by command over the second subchannel. The current dynamic dump operation is allowed to continue, following issuance of the two messages, unless the operator replies 'YES'.

Utility Action:
If the operator replies 'YES', the dump operation in progress over one subchannel is aborted by command over the second subchannel, and then a new dump operation is begun over the second subchannel. If the operator replies 'NO', the dynamic dump operation in progress is allowed to continue and the new dynamic dump request is cancelled.

User Action:
Determine from the individuals making the dynamic dump requests the priority of the requests, then respond 'YES' or 'NO' as appropriate.

IFL513I UNCONDITIONAL CONTINUE REJECTED BY EP-DYNADMP

Explanation:
After a 'YES' response to an IFL512A message, the host portion of the dynamic dump utility was unsuccessful in attempting to seize control of the EP portion of the utility.

Utility Action:
The host portion of the utility is abnormally terminated with a dump; this situation should not occur.

User Action:
Probable user error. Ensure that the parameters on the SYSUT1 DD statement are correct and resubmit the job.

IFL514I DYNADMP FUNCTIONS TERMINATED BY OPERATOR

Explanation:
The operator replied 'NO' to an IFL512A message. Consequently, the only dynamic dump commands that the operator may enter during the remainder of the current dynamic dump operation are formatting commands such as PRINT.

Utility Action:
None.

User Action:
None.

IFL600I xxxxxxxx NOT OPENED.

Explanation:
The named data set (SYSUT2, SYSPRINT, or SYSIN) could not be opened. Either the DD statement defining the data set was not included in the input stream or a DCB parameter was found invalid.

Utility Action:
The function is terminated.

User Action:
Probable user error. Ensure that the DD statements for SYSUT2, SYSPRINT, and SYSIN are included in the input stream and that the parameters on the DD statement are correct. Resubmit the job.

Appendix F: DOS Utility Messages

This appendix shows the format of each error, warning, and completion message issued to the programmer or to the operator of the host processor CPU during execution of the loader, dump, and dynamic utility programs under the control of the disk operating system.

Messages Issued by the Loader

```
IFU000I  3704/3705 LOAD COMPLETE 3704/3705-xxx LOADMOD=(file name).
```

Explanation:
The controller whose address is *xxx* was successfully loaded with the emulation program load module indicated by (*file name*).

Utility Action:
The loader utility processes the next control card, if any.

User Action:
None.

```
IFU001I  ERROR--LOADING PROCESS TERMINATED ** CONTROL STATEMENT ERROR **
```

Explanation:
The LOAD control card contained a syntax error.

Utility Action:
The controller is not loaded; the loader utility processes the next utility control card, if any.

User Action:
Correct the control card and resubmit the job.

```
IFU002I  ERROR--LOADING PROCESS TERMINATED ** -MISSING KEYWORD- **
```

Explanation:
A required keyword parameter is missing from the LOAD utility control card.

Utility Action:
The loader utility processes the next utility control card, if any.

User Action:
Correct the control card and resubmit the job stage.

```
IFU003I  ERROR--LOADING PROCESS TERMINATED ** xxx PERMANENT I/O ERROR **
```

Explanation:
A permanent I/O error occurred in the communications controller during loading.

Utility Action:
Loading of the controller is terminated, the loader utility processes the next utility control card, if any.

User Action:
Resubmit the loader job. If the I/O error persists, keep the listings and call IBM.

```
IFU004I  ERROR--LOADING PROCESS TERMINATED
         ** INITIAL-TEST DETECTED A HARDWARE FAILURE 3704/3705-xxx **
```

Explanation:
The initial test routine did not return control to the loader utility, thus indicating that a hardware error was detected in the communications controller.

Utility Action:
The controller is not loaded; the loader utility processes the next utility control card, if any.

User Action:
The operator may follow the problem determination procedure.

```
IFU005I  ERROR--LOADING PROCESS TERMINATED
         ** -INVALID 3704/3705 SYMBOLIC ADDRESS- **
```

Explanation:
The symbolic address specified in the 3705= parameter of the LOAD card is invalid.

Utility Action:
The controller is not loaded; the loader utility processes the next utility control card, if any.

User Action:
Change the LOAD control card and the associated ASSGN statement to a valid symbolic address, then resubmit the job.

```
IFU006I  UTILITY END xx WAS THE HIGHEST SEVERITY CODE.
```

Explanation:
The loader utility has processed all of the control cards in the input file. The possible values for *xx* are as follows:

00 The loading process was completed successfully; all controllers that were to be loaded are now loaded.

04 The loading process for at least one of the controllers to be loaded generated a warning or error message; the controller was not loaded.

08 Because of a severe error, none of the controllers to be loaded was successfully loaded.

Utility Action:
The loader job is completed.

User Action:
If the severity code is greater than 00, examine the message file for the appropriate messages. Correct the job control statements in error, and resubmit the job. Any load procedures that were successfully completed do not need to be resubmitted.

```
IFU007I  ERROR--LOADING PROCESS TERMINATED
         ** (file name) UNRECOVERABLE I/O ERROR **
```

Explanation:
A permanent I/O error was encountered when the loader utility attempted to read from the file specified by (*file name*).

Utility Action:
Loading of the controller is terminated; the loader utility processes the next utility control card.

User Action:
Ensure that LOADMOD specifies the correct file. If the correct file is already specified, resubmit the job. If *file name* is DIAGFLE, use DIAG=NO to bypass the problem. If the error persists, keep the listing and call IBM.

```
IFU008I  ERROR--LOADING PROCESS TERMINATED ** 3704/3705 DEVICE TYPE CONFLICT **
```

Explanation:
The specified device is not identified as a 3704 or 3705 to the operating system.

Utility Action:
The loader utility processes the next utility control card, if any.

User Action:
Ensure that the symbolic address specified in the 3705= parameter of the LOAD control card corresponds to the proper communications controller and that the DIAG= parameter was specified correctly.

```
IFU009I  ERROR-LOADING PROCESS TERMINATED ** DIAGFLE IN ERROR **
```

Explanation:
The diagnostic file (DIAGFLE) did not contain the required initial test routines in the proper sequence.

Utility Action:
The loader utility processes the next utility control card, if any.

User Action:
Ensure that the proper initial test file is specified in the DIAGFLE DLBL statement and that IFU3705D and IFU3705E were specified in that sequence.

```
IFU010I  3704/3705 LOAD FAILED 3704/3705-xxx LOADMOD=(file name).
```

Explanation:
The loading of the controller indicated by *xxx* failed. This message is sent only to the operator.

Utility Action:
The loader utility processes the next utility control card; if any.

User Action:
Examine the SYSOUT output for messages defining the problem and respond accordingly.

Messages Issued by the Dump Program

IFU100D 3704/3705-xxx ACTIVE *** REPLY TO CONTINUE ***

Explanation:
An attempt has been made to dump the controller indicated by *xxx* but this controller is currently executing a program. This message is sent only to the operator.

Utility Action:
The operator's reply is awaited.

User Action:
If dumping of the controller is desired (thereby terminating execution of its program), the operator should respond YES to this message. Otherwise, he should respond NO or cancel the job.

IFU101I INVALID CONTROL STATEMENT; DEFAULT TAKEN.

Explanation:
The DUMP statement is incorrectly coded.

Utility Action:
The dump utility provides an unformatted dump of the entire contents of the communications controller.

User Action:
Probable user error. If the dump that is provided does not provide sufficient information, correct the dump statement and resubmit the job.

IFU103I 3704/3705-xxx COULD NOT BE DUMPED.

Explanation:
An error has occurred, and none of the contents of the controller could be copied to the work file.

Utility Action:
The dump job is terminated.

User Action:
Ensure that the data set characteristics accurately describe the data set being used. If they do not, correct the characteristics and resubmit the job. If the error persists, optionally follow the problem determination procedure.

IFU104I 3704/3705-xxx HAS BEEN DUMPED SUCCESSFULLY.

Explanation:
The contents of the 3704 or 3705 whose unit address is *xxx* have been transmitted to the work file (SYS008).

Utility Action:
The dump utility program will next interpret the dump control cards and produce the requested outputs.

User Action:
None.

IFU105I CONTROL STATEMENT MISSING; DEFAULT TAKEN.

Explanation:
No DUMP control statement was provided.

Utility Action:
A full storage dump is produced.

User Action:
None.

IFU500I SYS011 NOT OPENED.

Explanation:
SYS011 could not be opened. The ASSGN statement defining the controller was not included in the input stream.

Utility Action:
The dynamic trace function is terminated.

User Action:
Probable user error. Ensure that the ASSGN statement for SYS011 is included in the input stream and that the parameters on the ASSGN statement are correct. Resubmit the job.

IFU501A REPLY WITH DESIRED FUNCTION OR END.

Explanation:
The dynamic dump utility is requesting control statement input from the operator because there was no input stream, or because a PAUSE statement was encountered in the input stream.

Utility Action:
The utility waits for the operator's reply.

User Action:
Enter a control statement. If all desired functions are complete, enter END. If more control statements reside in the input stream, enter SYSIN.

IFU502I THE FUNCTION COULD NOT BE PERFORMED.

Explanation:
A permanent input/output error was encountered while processing SYS011.

Utility Action:
The utility terminates the job step with a dump.

User Action:
Ensure that DYNADMP=YES is specified in the BUILD macro, and that the unit allocated to SYS011 is the type 1 channel adapter of the communications controller.

IFU503I FUNCTION COMPLETED - xx.

Explanation:
The function has completed with the value indicated by *xx*.

Utility Action:
The function is terminated.

User Action:
If *xx* is 00, completion is normal and no action is required. If *xx* is 08, ensure that the control statements appear in the proper sequence. The function was not acceptable to the emulation program.

IFU504I INVALID CONTROL STATEMENT.

Explanation:
A control statement was incorrectly specified.

Utility Action:
The function is terminated.

User Action:
Probable user error. Ensure that the control statement is valid and retry the function.

IFU505A OC EXIT TO FORCE STOP.

Explanation:
A dynamic trace dump has been requested. The dump may be stopped by establishing operator communication.

Utility Action:
The dynamic dump is terminated when operator communication is established.

User Action:
When it is determined that enough trace information has been dumped, establish operator communication for a background partition by pressing the console interrupt button, or for foreground partition Fx, reply `'MSGFx'` in response to the attention routine. This ends the dynamic dump when the current trace table entry is complete. The dump can be stopped only when there is activity on the line being traced. If there is no current activity on any line being traced, start a trace on some line on which there is activity; the trace may be started from the control panel of the controller. If no line being traced has any activity, the trace can be stopped only by canceling the dynamic dump utility and reloading the communications controller.

IFU506I FORCE STOP ACKNOWLEDGED.

Explanation:
The dynamic dump utility has received the STOP indication.

Utility Action:
The utility stops the dump.

User Action:
None

IEU507I FUNCTION NOT AVAILABLE OR INVALID.

Explanation:
The desired function was not available due to unavailable devices or was found invalid by the controller portion of the dynamic dump utility.

Utility Action:
The function is terminated.

User Action
Probable user error. Ensure that the control statement is valid and that required devices are available, then retry the function.

IFU508I TRACE BLOCK nnn,nnn WRITTEN AT hh:mm:ss.

Explanation:
Trace block number *nnn,nnn* was written to the work file at time *hh:mm:ss* (hour:minute:second).

Utility Action:
All trace blocks are time-stamped before being written to the work file. For the first trace block, for the last trace block, and for every 200th trace block in-between, the message IFU508I is sent to the host processor console.

User Action:
Refer to the IFU508I messages when printing the work file. The PRINT facility permits selective printing according to time.

IFU510I NO TRACE BLOCKS SATISFY THE PRINT COMMAND.

Explanation:
No trace blocks were found written to the work file after the time specified (START=) in the PRINT control statement.

Utility Action:
The dynamic dump utility does not print any trace blocks; however, if there is data other than trace blocks on the work file (for example, storage dumps) this nontrace data will be printed.

User Action:
Refer to the IFU508I messages sent to the operator's console during the time the trace was taken. These will indicate the range of time stamps associated with the trace blocks written to the work file. From this information, derive a meaningful START time to specify on the PRINT command.

IFU511I EP-DYNADMP IS UNDER THE CONTROL OF ANOTHER HOST

Explanation:
The EP portion of the dynamic dump utility is busy with the dynamic dump requests of another host processor.

Utility Action:
Message IFU512A is issued.

User Action:
None.

IFU512I REPLY 'YES' TO CONTINUE UNCONDITIONALLY OR 'NO' TO END

Explanation:
This message follows message IFU511I and requires an operator response to indicate the utility action desired. The emulation program can transfer dynamic dump data over only one subchannel at a time. Therefore, a dynamic dump operation over one subchannel must end before a dynamic dump operation over a second subchannel can begin. The operation over the first subchannel can end normally or it can be aborted by command over the second subchannel. The current dynamic dump operation is allowed to continue, following issuance of the two messages, unless the operator replies 'YES'.

Utility Action:
If the operator replies 'YES', the dump operation in progress over one subchannel is aborted by command over the second subchannel, and then a new dump operation is begun over the second subchannel. If the operator replies 'NO', the dynamic dump operation in progress is allowed to continue and the new dynamic dump request is cancelled.

User Action:
Determine from the individuals making the dynamic dump requests the priority of the requests, then respond 'YES' or 'NO' as appropriate.

IFU513I UNCONDITIONAL CONTINUE REJECTED BY EP-DYNADMP

Explanation:
After a 'YES' response to an IFU512A message, the host portion of the dynamic dump utility was unsuccessful in attempting to seize control of the EP portion of the utility.

Utility Action:
The host portion of the utility is abnormally terminated with a dump; this situation should not occur.

User Action:
Probable user error. Ensure that the parameters on the SYSUT1 DD statement are correct and resubmit the job.

IFU514I DYNADMP FUNCTIONS TERMINATED BY OPERATOR

Explanation:
The operator replies 'NO' to an IFU512A message. Consequently, the only dynamic dump commands that the operator may enter during the remainder of the current dynamic dump operation are formatting commands such as PRINT.

Utility Action:
None.

User Action:
None.

IFU600I SYS010 NOT OPENED

Explanation:
The named file, SYS010, could not be opened. The ASSIGN statement specified 'ignore'.

Utility Action:
The function is terminated.

User Action:
Probable user error. Ensure that the ASSGN statement for SYS010 is correct. Resubmit the job.

Appendix G: Transmission Codes for World Trade Teletypewriter Terminals

The chart below gives the eight-bit representation within communications controller storage of the transmission code bit patterns for each character in the ITA2 and ZSC3 character sets, as used by various European teleprinters (World Trade teletypewriters). In the WTTYEOB and WTTYEOT operands of the GROUP macro, if used, specify hexadecimal values appearing within the chart.

Note: The transmission code bit pattern, as it appears on the communication line, is the reverse of the value shown in this chart.

Character			Transmission Code	
Letters shift	Figures shift		Letters shift	Figures shift
ITA2 & ZSC3	ITA2	ZSC3	(hexadecimal)	(hexadecimal)
A	-	+	03	23
B	?	6	19	39
C	:	8	0E	2E
D	WRU	WRU	09	29
E	3	-	01	21
F		4	0D	2D
G		0	1A	3A
H		?	14	34
I	8	Bell	06	26
J	Bell	2	0B	2B
K	(	(	0F	2F
L	)	)	12	32
M	.	7	1C	3C
N	,	,	0C	2C
O	9	:	18	38
P	0	9	16	36
Q	1		17	37
R	4	/	0A	2A
S	'	'	05	25
T	5	.	10	30
U	7	1	07	27
V	=	=	1E	3E
W	2	3	13	33
X	/		1D	3D
Y	6	5	15	35
Z	+		11	31
CR	CR	CR	08	28
LF	LF	LF	02	22
LTRS	LTRS	LTRS	1F	-
FIGS	FIGS	FIGS	1B	-
Space	Space	Space	04	24

Appendix H: Procedure for Determining Line Interrupt Priorities

This appendix gives a recommended procedure for determining the interrupt priorities for lines serviced by type 2 and type 3 communication scanners.

Optimum servicing of communication lines and maximum protection from overruns are achieved when (1) the priority registers associated with all four interrupt priorities are used and (2) the total throughput of the network in bits per second is about evenly distributed among the four different priorities.

The following procedure ensures that the four interrupt priorities are evenly distributed among lines in the network.

1. Divide by 8 the speed (as specified by the SPEED operand of the LINE macro) of each line serviced by a *type 3* communication scanner. The result is the adjusted line speed for such lines and is to be used in the remaining steps of the procedure. For lines serviced by a *type 2* scanner the adjusted line speed is the same as the speed specified in the SPEED operand.
2. For each speed category, multiply the adjusted speed by the number of lines to which that speed applies.
3. Calculate the total throughput rate for all lines (in bits per second) by adding up the values calculated in step 2. Then divide the result by 4 to determine one-fourth of the throughput in bits per second.
4. List all lines in the network in the sequence of their adjusted line speeds. Those lines having the highest adjusted speed should appear at the top of the list and those having the lowest adjusted speed should appear at the end. Within any speed category the sequence of lines does not matter.
5. Divide the list of lines into four sections such that the total throughput for each section is roughly the same as one-fourth of the total throughput.
6. Assign priority 3 to each line in the first section of the list, priority 2 to the lines in the second section, priority 1 to those in the third section, and priority 0 to those lines in the last section.
7. Specify the assigned priority in the INTPRI operand of the LINE macro representing each line.

For purposes of the foregoing procedure, approximately the same proportion of lines in each section of the list are assumed to be active at any given moment. If the planned use of the network or experience shows that the proportions are markedly different, you may wish to adjust the distribution of lines to the sections of the list to compensate, then respecify the values in the INTPRI operands for the affected lines.

For instance, if experience shows that several of the lines in the last section of the list are relatively inactive compared to lines in the other sections, you could adjust each of the section boundaries upward so that more lines appear in the last group and fewer in each of the other sections.

The use of the procedure is illustrated by the following examples.

Example One:

The network has six lines rated at 9,600 bps and serviced by a type 3 scanner, and 13 lines—7 rated at 2,400 bps and 6 rated at 600 bps—serviced by a type 2 scanner.

Determine the total throughput and one-fourth of the total:

Line ID (Name of LINE Macro)	Line Speed (LINE: SPEED)	Type of Scanner	Adjusted Line Speed	Number of Lines	Throughput
			(Step 1)		(Step 2)
LH1-LH6	9,600	3	1,200	6	7,200
LM1-LM7	2,400	2	2,400	7	16,800
LL1-LL6	600	2	600	6	3,600
		Total throughput (Step 3)			27,600
		One-fourth of total			6,900

List lines in order of adjusted speed (step 4), divide list into four sections (step 5), and assign priorities to each section (step 6):

Line ID (Name of LINE Macro)	Adjusted Line Speed	Total Bit Rate for Section	Interrupt Priority
LM1	2,400	7,200	3
LM2	2,400		
LM3	2,400		
LM4	2,400		
LM5	2,400	7,200	2
LM6	2,400		
LM7	2,400		
LH1	1,200	6,000	1
LH2	1,200		
LH3	1,200		
LH4	1,200		
LH5	1,200		
LH6	1,200		
LL1	600		
LL2	600	7,200	0
LL3	600		
LL4	600		
LL5	600		
LL6	600		

As step 7, specify the priority values in the INTPRI operands of the LINE macros.

Example Two:

The network has 11 lines serviced by a type 3 scanner—three lines at 19,200 bps, three at 9,600 bps, and five at 2,400—and the following lines serviced by a type 2 scanner: ten lines at 1,200 bps, nine at 600 bps, seven at 150 bps, and eight at 134.5 bps.

As in example one, first determine the total throughput and one-fourth of that value, then list the lines in order of adjusted line speed, divide into four sections, and assign priorities.

Line ID (Name of LINE Macro)	Line Speed (LINE: SPEED)	Type of Scanner	Adjusted Line Speed	Number of Lines	Throughput
LA1-LA3	19,200	3	2,400	3	7,200
LB1-LB3	9,600	3	1,200	3	3,600
LC1-LC5	2,400	3	300	5	1,500
LD1-LD10	1,200	2	1,200	10	12,000
LE1-LE9	600	2	600	9	5,400
LF1-LF7	150	2	150	7	1,050
LG1-LG8	134.5	2	134*	8	1,072

*Decimal fraction dropped as insignificant

Total throughput	31,822
One-fourth of total	7,956

Line ID (Name of LINE Macro)	Adjusted Line Speed	Total Bit Rate for Section	Interrupt Priority
LA1	2,400		
LA2	2,400	7,200	3
LA3	2,400		
LB1	1,200		
LB2	1,200		
LB3	1,200		
LD1	1,200	8,400	2
LD2	1,200		
LD3	1,200		
LD4	1,200		
LD5	1,200		
LD6	1,200		
LD7	1,200		
LD8	1,200	8,400	1
LD9	1,200		
LD10	1,200		
LE1	600		
LE2	600		
LE3	600		
LE4	600		
LE5	600		
LE6	600		
LE7	600		
LE8	600		
LE9	600		
LC1	300		
LC2	300		
LC3	300		

Line ID (Name of LINE Macro)	Adjusted Line Speed	Total Bit Rate for Section	Interrupt Priority
LC4	300		
LC5	300		
LF1	150		
LF2	150	7,822	0
LF3	150		
LF4	150		
LF5	150		
LF6	150		
LF7	150		
LG1	134		
LG2	134		
LG3	134		
LG4	134		
LG5	134		
LG6	134		
LG7	134		
LG8	134		

Appendix I: Upper Scan Limits, Address Substitution, and High Speed Select Options

The maximum data rate, or speed, at which a communication line can operate is limited by the frequency at which that line's interface address is scanned by the communication scanner. In the absence of upper scan limits, address substitution and high speed select options, each line interface address associated with a type 2 or type 3 communication scanner is scanned once per scanning cycle. The maximum line speed in this case is 4800 bits per second (bps). (The type of line set and oscillator or modem clocking rate determines the actual line speed.) To accommodate higher maximum line speeds requires the imposition of upper scan limits or the application of the address substitution or high speed select technique.

Upper Scan Limits

Imposing an upper scan limit is a means of increasing the frequency at which a selected range of line addresses is scanned, at the expense of not scanning the remaining addresses associated with the scanner. (No lines attached to the unscanned addresses can be active while the upper scan limit is in effect.) By not scanning some addresses, the communication scanner can scan the others more often within each scanning cycle, thus raising the maximum line speed. (The number of scans per cycle is constant regardless of how they are distributed to the line addresses.)

Upper scan limits can be specified individually for each of the type 2 and type 3 scanners in the communications controller. Figure I-1 shows for each scan limit value (from 0 to 3, as specified in the GENEND macro of the emulation program or network control program), the range of addresses that are scanned (light boxes) and those not scanned (dark boxes). Also given is the maximum line speed for the addresses scanned (disregarding any lower limit that may be imposed by choice of line set and oscillator speeds).

Address Substitution

A technique similar to the use of upper scan limits alters the scanning pattern so that a single, predetermined address is scanned several times per scanning cycle, again at the expense of not scanning other addresses. In this technique, however, only one address, of the range of addresses scanned, receives the benefit of increased scanning frequency. Further, address substitution applies uniformly to *all* installed type 2 and type 3 scanners, rather than being specified separately for the individual scanners. Address substitution should not be specified if the controller contains a type 3 scanner.

Figure I-2 shows, for each bit position in the address substitution mask (specified in the GENEND macro of the emulation program or network control program), the selected addresses that will be scanned more often, and the group of addresses that will accordingly not be scanned. As is the case for upper scan limits, no lines attached to the unscanned addresses can be active while address substitution is in effect.

High Speed Select Option

The high speed select option is similiar to address substitution in that bit settings within a mask alter the scanning pattern so that a predetermined address is scanned several times per scanning cycle, at the expense of not scanning other

addresses. This option differs from address substitution, however, in that (1) up to eight addresses serviced by the scanner can receive the increased scanning frequency, and (2) masks are individually specified for each of the installed scanners, thus allowing more flexibility in selecting addresses to receive the increased scanning. Figure I-3 shows, for each bit position in the high speed select mask (specified in the GENEND macro) the selected addresses that will be scanned more often and the group of addresses that will not therefore be scanned. No lines attached to the unscanned addresses can be active while the high speed select mask is in effect.

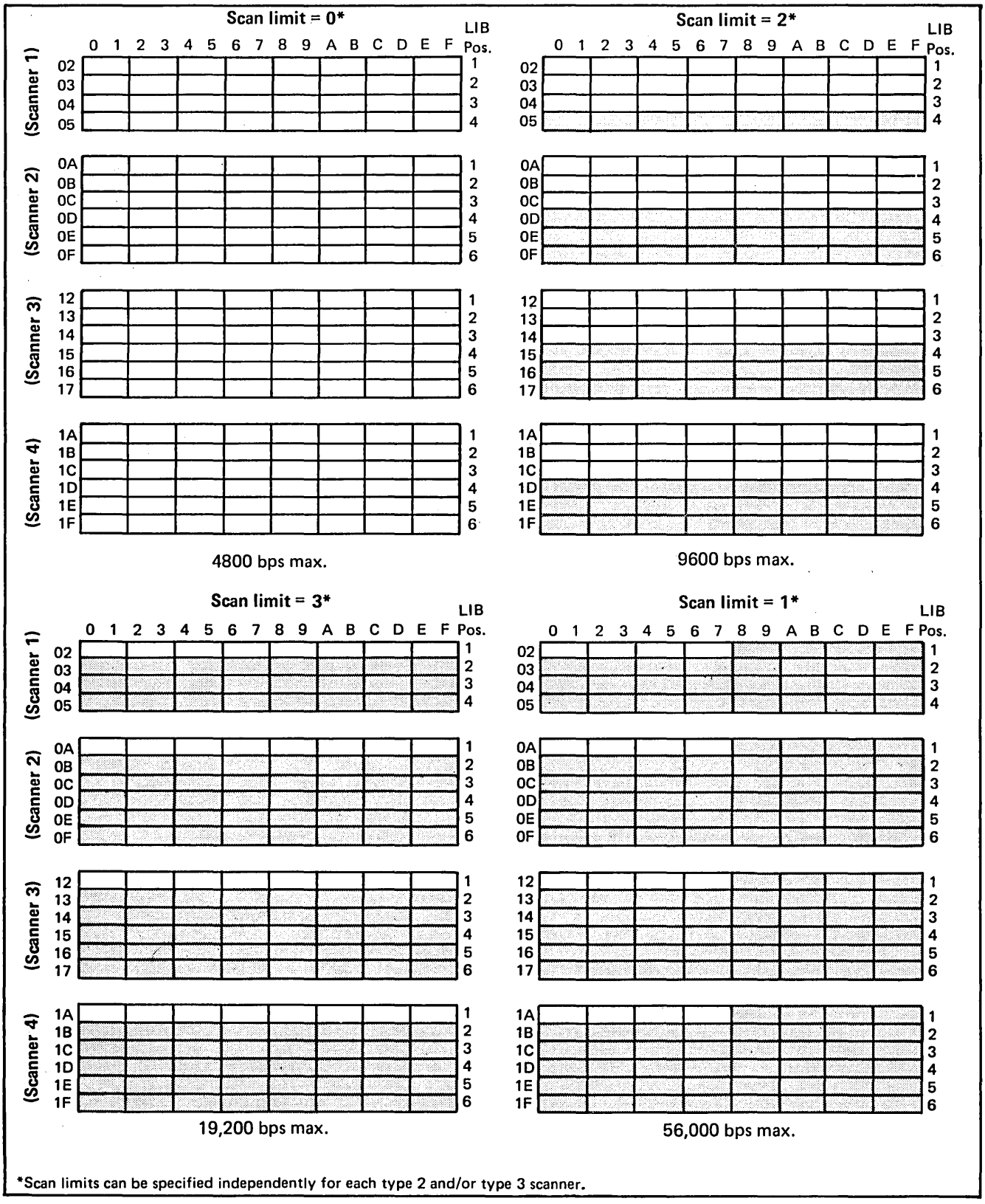

Figure I-1. Addresses Scanned and Not Scanned when Upper Scan Limits are Used

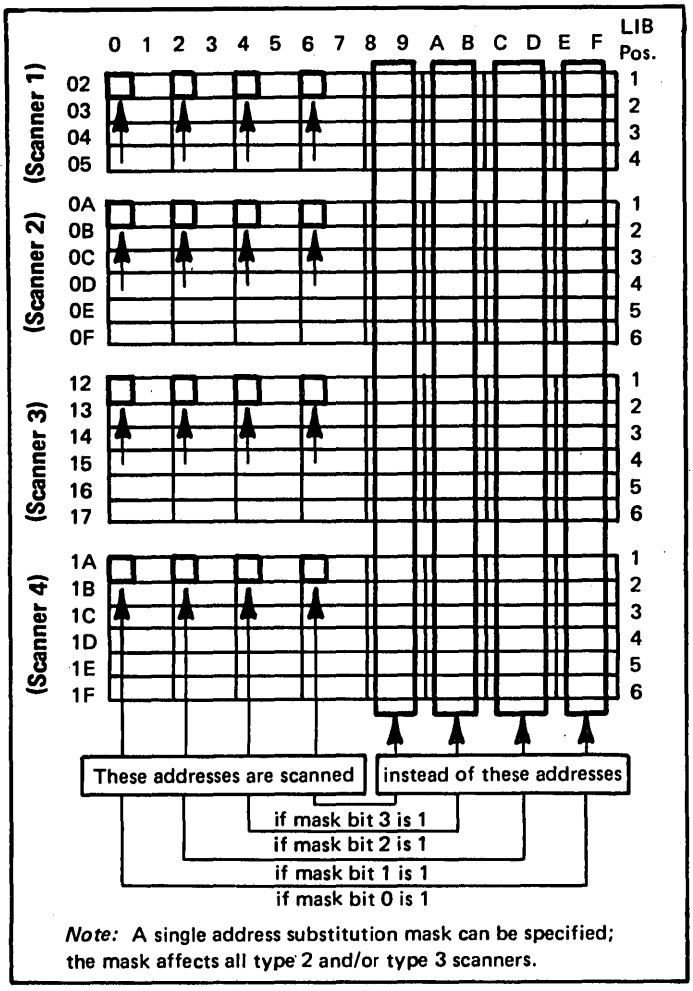

Figure I-2. Addresses Scanned and Not Scanned when Address Substitution Mask is Used

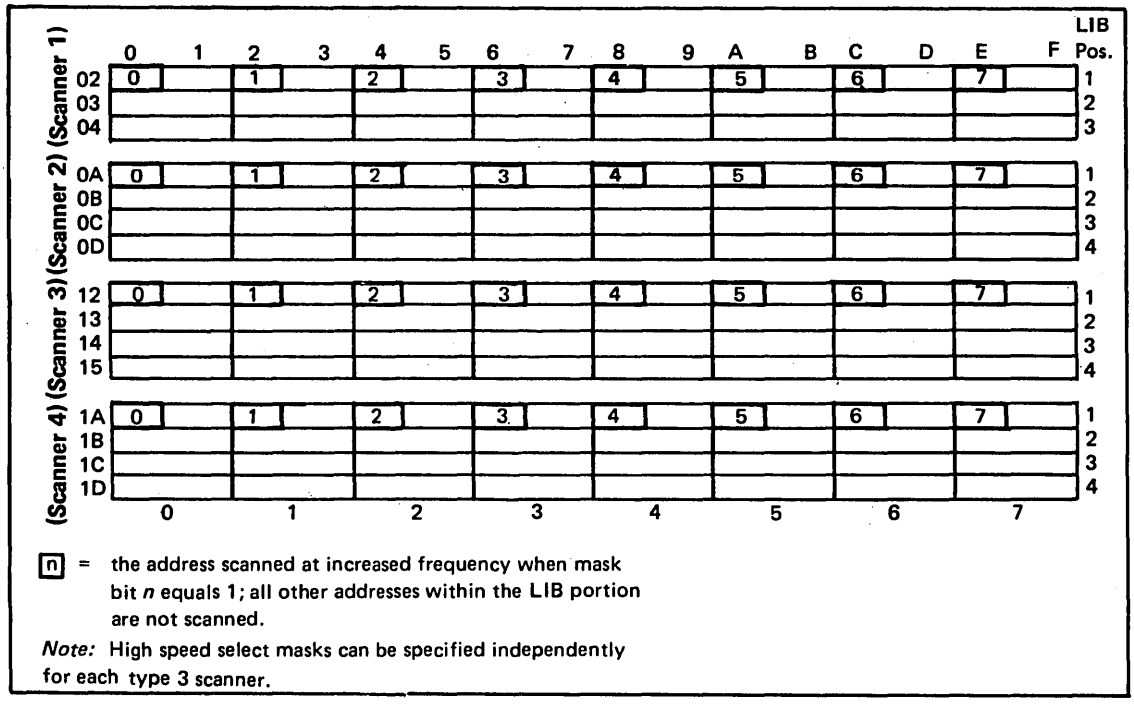

Figure I-3. Addresses Scanned and Not Scanned when High Speed Select Mask is Used

Appendix J: Supplemental Information for Airlines Line Control Users

This appendix contains information needed to specify communication lines to use the airlines line control (ALC) procedure. The information herein, which supplements the information in the remainder of this publication, is applicable only when you are defining a control program for an IBM 3705-II Communications Controller in which RPQ numbers 858911 and 858912 are installed.

To define a line as using airlines line control (ALC), rather than BSC or start-stop line control, specify LNCTL=ALC in the GROUP macro and observe the requirements indicated under each of the macro instructions and operands mentioned below.

Note: LNCTL=ALC is valid only for a line serviced by a type 3 communication scanner in a 3705-II controller.

Certain operands of the LINE macro may be specified instead in the GROUP macro. The operands for which you may do so are the same for an ALC line as for a non-ALC line

BUILD Macro Instruction

Use of ALC imposes no special requirements on coding of the BUILD macro and its operands.

CSB Macro Instruction

No line for which LNCTL=ALC is specified should be specified in the WRAPLN operand of the CSB macro unless all lines serviced by the scanner represented by the CSB macro are ALC lines. Otherwise, use of ALC imposes no special requirements on coding of the CSB macro and its operands.

GROUP Macro Instruction

Each ALC line must be included in a line group, represented by a GROUP macro. No ALC line may be included in more than one line group, and all lines in an ALC line group must be ALC lines.

All ALC lines are nonswitched (DIAL=NO).

In general, the operands of the GROUP macro that apply to nonswitched BSC lines apply also to ALC lines. Specific requirements for coding these operands are as follows.

Invalid Operands

CHAREC	QUIETCT
DELAY	REPLYTO
EOB	TEXTTO
EOT	

Valid Operands

The following operands are valid for an ALC line group. Specific requirements are indicated.

DIAL:	Specify DIAL=NO or omit this operand.
LNCTL:	Specify LNCTL=ALC.

In general, the operands of the LINE macro that apply to nonswitched BSC lines apply also to ALC lines. Specific requirements for coding these operands are as follows.

Invalid Operands

The following operands are invalid for ALC lines and if coded are ignored.

AUTO	QUIET
CHECK	RING
CODE	TADDR
DUALCOM	TERM
MULTI	UNITXC

Valid Operands

The following operands are valid for an ALC line. Specific requirements are indicated.

ADDRESS and RCVADDR: ALC lines are arranged in pairs and attached to consecutive line interface addresses, the lower of which must be an even address. The even address is used for transmit operations. The next higher (odd) address is used for receive operations. In the ADDRESS operand specify only the even (transmit) address and the associated emulation subchannel address. *Example:* ADDRESS=(0A0,43). In the RCVADDR operand specify the odd (receive) address and the associated emulation subchannel address. *Example:* RCVADDR=(0A1,42).

The ADDRESS and RCVADDR operands are required.

Note: Any line interface address you specify for an ALC line must be an address associated with a type 3 communication scanner.

Specify the subchannel address in the ADDRESS and RCVADDR operands in the same way as for a non-ALC line. Because an ALC line must be associated with a type 4 channel adapter, you may specify multiple subchannel addresses for the line interface address you specify. (See Figure J-2 for an example.)

BUFSIZE: Specify BUFSIZE=(n1 ,[n2]), in which *n1* represents the emulation mode buffer size for the transmit (even) line address specified in the ADDRESS operand and *n2* represents the emulation mode buffer size for the receive (odd) line address specified in the RCVADDR operand.

n1 and *n2* may be any of the following values:

8	64	160
16	96	192
32	128	224

If you omit this operand, a buffer size of 32 [BUFSIZE=(32,32)] is assumed.

CHNPRI: No special requirements apply to the use of this operand.

CLOCKNG:	No special requirements apply to the use of this operand.
CU:	Specify CU=2703 or omit this operand.
CUTYPE:	Specify CUTYPE=1006 or CUTYPE=2946 or CUTYPE=2948, as appropriate for the type of station attached to the ALC line. If a mixture of station types are attached to the line, specify CUTYPE=MIXD.
DATRATE:	No special requirements apply to the use of this operand.
DISABLE:	Specify DISABLE=NO or omit this operand.
DUPLEX:	Specify DUPLEX=FULL.
INTPRI:	No special requirements apply to the use of this operand.
MODEM:	Specify MODEM=OPTION2 or omit this operand.
NEWSYNC:	No special requirements apply to the use of this operand.
PAD:	Specify PAD=YES or omit this operand.
RCVADDR:	(See ADDRESS operand in this appendix.)
SPEED:	Line speeds between 2400 and 9600 bps, inclusive, are valid.

GENEND Macro Instruction

Use of ALC imposes no special requirements on coding of the GENEND macro and its operands.

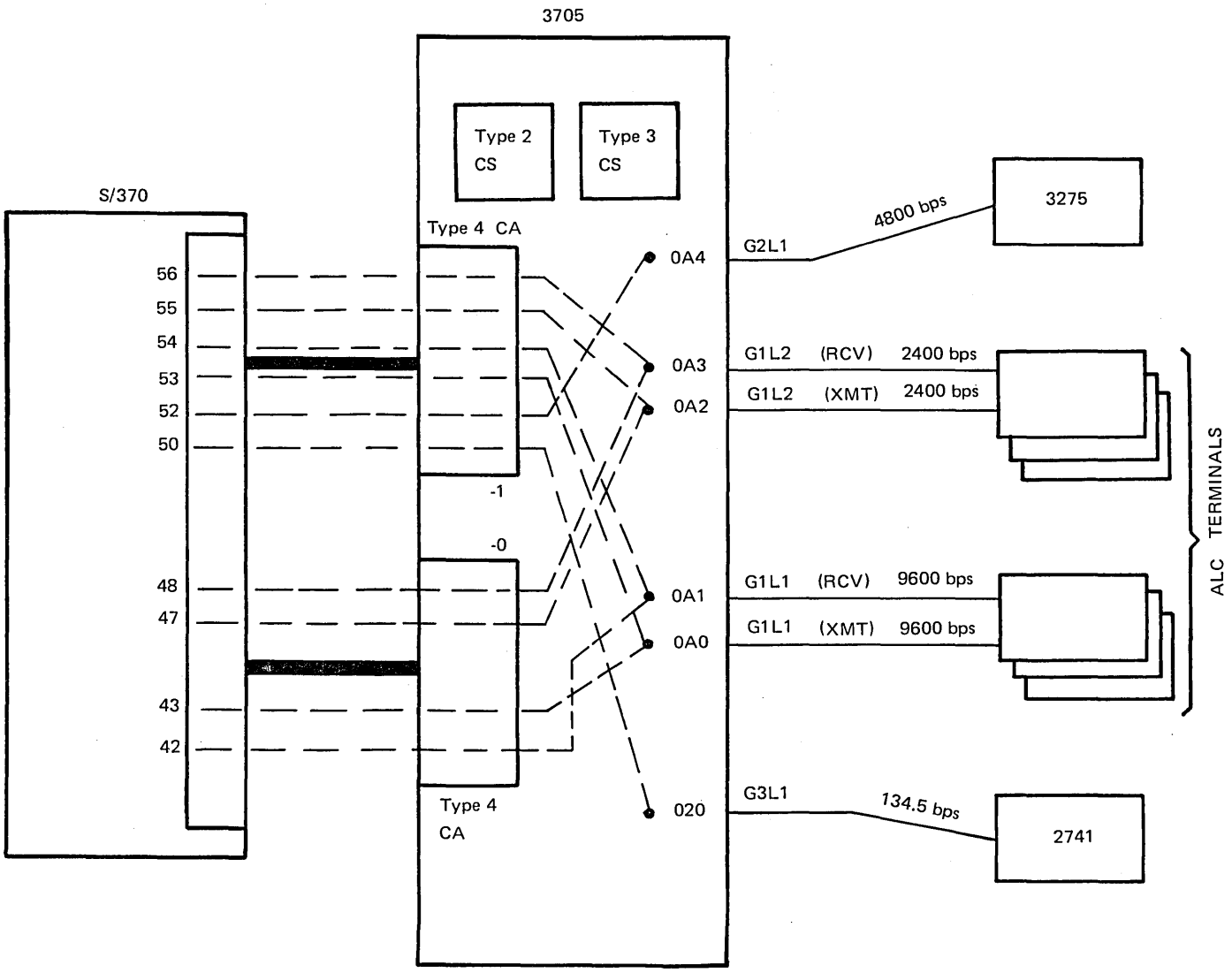

Figure J-1. Network Configuration for Sample Program for ALC Users'

```
         BUILD    MODEL=3705-2,                                    X
                  CA=(TYPE4, TYPE4),                               X
                  LOCHAN=(40, 50),                                 X
                  HICHAN=(4F, 5F),                                 X
                  TEST=YES,                                        X
                  LINETRC=YES,                                     X
                  DYNADMP=(YES,, 5F),                              X
                  LOADLIB=LOD3705,                                 X
                  OBJLIB=OBJ3705,                                  X
                  NEWNAME=ALC01

         CSB      TYPE=TYPE2,                                      X
                  MOD=0,                                           X
                  SPEED=(134, 600),                                X
                  WRAPLN=020

         CSB      TYPE=TYPE3,                                      X
                  MOD=1,                                           X
                  SPEED=(150, 600, 1200),                          X
                  WRAPLN=0A4

G1       GROUP    LNCTL=ALC,                                       X
                  CLOCKNG=EXT,                                     X
                  DUPLEX=FULL,                                     X
                  CUTYPE=MIXD

G1L1     LINE     ADDRESS=(0A0, 43, 53-1),                         X
                  RCVADDR=(0A1, 42, 54-1),                         X
                  SPEED=9600,                                      X
                  BUFSIZE=(64, 32),                                X
                  INTPRI=3,                                        X
                  CHNPRI=HIGH

G1L2     LINE     ADDRESS=(0A2, 47, 55-1),                         X
                  RCVADDR=(0A3, 48, 56-1),                         X
                  SPEED=2400

G2       GROUP    LNCTL=BSC,                                       X
                  CODE=EBCDIC

G2L1     LINE     ADDRESS=(0A4, 52-1),                             X
                  SPEED=4800,                                      X
                  TERM=3275,                                       X
                  CU=2701

G3       GROUP    LNCTL=SS,                                        X
                  CLOCKNG=INT

G3L1     LINE     ADDRESS=(020, 50-1),                             X
                  SPEED=134,                                       X
                  DUPLEX=FULL,                                     X
                  UNITXC=NO,                                       X
                  TERM=2741

         GENEND
         END
```

Figure J-2. Sample Emulation Program for ALC Users

Glossary

access method: A data management technique for transferring data between main storage and an input/output device. In this publication, teleprocessing access method refers to the data management technique executed in the host processor that transfers data between the host processor and the emulation program in the communications controller.

address substitution: A feature for type 2 communication scanners that modifies the scan counter output to replace certain pairs of interface addresses with one other address.

addressing: The means whereby the originator or control unit selects the teleprocessing device to which it is going to send a message.

block: The smallest data unit recognized by the communications controller. For start-stop devices, a unit of data between two EOB characters; for BSC devices, a unit between two ETB or ETX characters.

central control unit: The communications controller hardware unit that contains the circuits and data flow paths needed to execute the controller's instruction set and to control storage and the attached adapters.

channel adapter: A communications controller hardware unit that provides attachment of the communications controller to a System/360 or System/370 channel.

communication scanner: A communications controller hardware unit that provides the interface between line interface bases and the central control unit. The communication scanner monitors the communication lines for service requests.

conditional operand: An operand of an emulation generation macro instruction that must be coded or omitted depending on whether certain other operands are coded or omitted.

configuration macro: One of the emulation generation macros that provide information necessary to construct the tables needed by the emulation program to control the flow of data between the communications controller and stations and between the communications controller and the host processor.

dump program: A utility program operating partly in the host processor and partly in the communications controller that (1) transfers the entire contents of controller storage to the host processor and (2) transfers user-selected portions of the contents to an output data set.

dynamic dump utility: A utility operating partly in the emulation program and partly in the host processor that (1) transfers the entire contents of the controller to the host, or (2) transfers the entire contents of the trace table to the host, or (3) transfers the trace table entries, as they are made, to the host processor. The controller continues to operate under the control of the emulation program during and after the dumping process.

emulation program: An IBM-supplied control program for the communications controller.

emulation generation language: The set of macro instructions and associated operands by which the communications controller user defines the network configuration and operating parameters of the teleprocessing subsystem.

emulation generation procedure: A two-stage process that creates an emulation program load' module based on parameters specified by the user through the emulation generation language.

extended addressing: The addition of two high-order bits to the basic addressing scheme to permit installation of larger storage capacities.

generation delimiter macro: The emulation program macro that marks the end of the emulation program generation input stream.

host processor: The central processing unit to which the communications controller is attached by a channel and that executes the teleprocessing access method that supports the communications controller.

initial test routine: A diagnostic program executed in the communications controller before the emulation program is loaded. The initial test routine tests the communications controller hardware for conditions that might cause failure after operation begins.

interrupt: A halt in processing that allows processing to be resumed at the place it left off.

interrupt priority: The order in which emulation program processes interrupts received simultaneously from two or more communication lines.

line control character: A special character that controls transmission of data over a communication line. For example, line control characters are used to start or end a transmission, to cause transmission-error checking to be performed, and to indicate whether a station has data to send or is ready to receive data.

line group: A group of communication lines by which stations supported by the same line-control discipline are connected to the communications controller.

line interface base (LIB): A communications controller hardware unit that provides for the attachment of up to 16 communication lines to the communications controller.

load module: A program in a format suitable for loading into storage for execution. an emulation program load module is produced by the linkage editor during the emulation program generation procedure; the loader utility loads it into the communications controller.

loader program: A utility program operating partly in the host processor and partly in the communications controller, that transfers an emulation program load module from host processor storage to the communications controller.

message: For BSC devices, the data unit from the beginning of the transmission to the first ETX character, or between two ETX characters; for start-stop devices, *message* and *transmission* have the same meaning.

scan limit: A type 2 communication scanner feature that allows the control program to limit the number of lines that a particular scanner addresses.

station: A point in a teleprocessing network at which data can either enter or leave.

subchannel: The channel facility required for sustaining a single I/O operation.

teleprocessing device: A unit of teleprocessing equipment connected to the communications controller via a communication line and identified as a cluster, terminal, or component at emulation program generation time.

terminal: A teleprocessing device capable of transmitting or receiving data (or both) over a communications line.

trace table: An area within the emulation program into which line trace information is placed.

transmission code: The character code used for data transmissions across a communication line.

transmission control unit (TCU): A unit that provides the interface between communication lines and a computer. The TCU interleaves the transfer of data from many lines across the subchannels to the computer.

Index

IBM 3704 and 3705 Emulation Program
Generation and Utilities Guide and
Reference Manual

GC30-3002-7

READER'S COMMENT FORM

Your comments, accompanied by answers to the following questions, help us produce better publications for your use. Each reply is carefully reviewed by the persons responsible for writing and publishing this material. Comments and suggestions become property of IBM.

Please direct *any requests for copies of publications or for assistance in using your IBM System or equipment,* to your IBM representative or to the IBM sales office serving your locality.

- Did this publication meet your needs?
- What is your occupation?
- How did you use this publication?

COMMENTS

We would appreciate your comments: please give specific page and line references when appropriate. Additional comments may be made on the other side of this form.

If you would like a reply, complete the following (Please Print):

Your Name _____

Company Name _____

Department _____

Street Address _____

City _____ State _____ Zip Code _____

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

Cut or Fold Along This Line

ADDITIONAL COMMENTS:

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
Dept. E01
P.O. Box 12195
Research Triangle Park
North Carolina 27709

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)

Cut or Fold Along Line

IBM 3704 and 3705 Emulation Program Generation and Utilities (File No. S360/S370-30) Printed in U.S.A. GC30-3002-7

IBM

Technical Newsletter

This Newsletter No. GN30-3087
Date September 2, 1977

Base Publication No. GC30-3002-7
File No. S360/S370-30

Previous Newsletters GN30-3066

IBM 3704 and 3705
Emulation Program
Generation and Utilities
Guide and Reference Manual

This Technical Newsletter provides replacement and added pages for the subject publication. These pages remain in effect until the next edition of the publication. The pages to be replaced and added are:

iii-v
ix, x
3-17 – 3-22
3-43 – 3-44
3-45, (blank)
4-3 – 4-6
J-1 – J-5 (added)
X-1, X-2
X-3, (blank)

Each technical change is marked by a vertical line to the left of the change.

Summary of Amendments

This Technical Newsletter adds information needed by users of the IBM Airlines Control Program to specify communication lines on which airlines line control (ALC) is to be used. Also added is a new operand of the GROUP macro: QUIETCT.

Note: Please file this cover letter at the back of the manual to provide a record of changes.

IBM Corporation, Publications Center, Dept. E01, P.O. Box 12195, Research Triangle Park, N.C. 27709

Printed in U.S.A.

IBM

Technical Newsletter

This Newsletter No.	GN30-3066
Date	August 24, 1976
Base Publication No.	GC30-3002-7
File No.	S360/S370-30
Previous Newsletters	None

IBM 3704 and 3705
Emulation Program
Generation and Utilities
Guide and Reference Manual

This Technical Newsletter provides replacement pages for the subject publication. The replacement pages remain in effect until the next edition of the publication. The pages to be replaced are:

Title Page, Edition Notice

Summary of Amendments

The edition notice is corrected to indicate that the current edition of the manual applies to version 3, modification 0 of the Emulation Program (OS).

Note: Please file this cover letter at the back of the manual to provide a record of changes.

IBM Corporation, Publications Center, Dept. E01, P.O. Box 12195, Research Triangle Park, N.C. 27709

Printed in U.S.A.

IBM 3704 and 3705 Emulation Program Generation and Utilities (File No. S360/S370-30) Printed in U.S.A. GC30-3002-7

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)