For Planning Purposes Only

GC30-3000-0

Systems

IBM 3705 Communications Controller Network Control Program Generation and Utilities, Guide and Reference Manual

Program Number 360H–TX–034 (NCP)
360H–TX–035 (SSP)

IBM

First Edition (April 1972)

The material in this manual is preliminary and should be used for planning purposes only. Changes will be made to the information herein, and these changes will be reported in subsequent revisions or Technical Newsletters.

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

This manual has been prepared by the IBM Systems Development Division, Publications Center, Department E01, P. O. Box 12275, Research Triangle Park, North Carolina 27709. 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 the above address. Comments become the property of IBM.

Preface

This publication provides the information necessary to define and generate a network control program for the IBM 3705 Communications Controller, to load the program into the controller, and to dump the contents of the controller storage. The publication is directed to systems analysts and systems programmers responsible for defining and generating a network control program.

Note: *This is a* preliminary edition *and is to be used for* planning purposes only. *The information will remain preliminary until updated in a subsequent edition.*

You are expected to have a basic understanding of teleprocessing and teleprocessing access methods. You should also have a general knowledge of the purposes of the IBM 3705 Communications Controller; you may obtain this knowledge from the publication, *Introduction to the IBM 3705 Communications Controller*, (GA27-3051).

This publication is arranged as follows:

Chapter 1 introduces the network control program and summarizes the content of the book.

Chapter 2 describes the teleprocessing characteristics that the programmer must consider when defining the network control program.

Chapter 3 gives detailed descriptions of the macro instructions that define the network control program.

Chapter 4 describes the program generation procedure.

Chapters 5 and 6 explain how to use the Loader and Dump utility programs.

The appendixes list the teleprocessing devices with which the communications controller can communicate using the network control program, and list the messages produced by the program generation procedure and utility programs.

When first using this book, read chapter 1 for a description of the purposes of the network control program, the generation procedure, and the utility programs. Then use chapter 2 to familiarize yourself with the characteristics of a teleprocessing subsystem that will concern you when defining a network control program. Thereafter, you may use the detailed macro instruction descriptions in chapter 3 to define the network control program. (The symbols used in chapter 3 to describe how to code the macro instructions are explained at the beginning of the chapter under the heading *Macro Instruction Coding Conventions.)* The margin tabs in this chapter provide quick access to the individual macro instruction descriptions.

Contents

Figures

List of Abbreviations

BCD	Binary Coded Decimal
BH macro	block handling macro
BH set	block handler set
BHR	block handling routine
bksp	backspace
bps	bits per second
BSC	binary synchronous communications
CA	channel adapter
CPU	central processing unit
DOS	Disk Operating System
EBCD	Extended Binary Coded Decimal
EBCDIC	Extended Binary Coded Decimal Interchange Code
EIA	Electronic Industries Association
EOB	end of block
EOT	end of transmission
EP	Emulator Program
ERP	error recovery procedure
ETB	end of transmission block
ETX	end of text
ID	identification
I/O	input/output
IPL	initial program load
K	thousand (1,024, when referring to bytes of storage)
LIB	line interface base
OLTT	on-line terminal test
OS	Operating System
RPQ	Request for Price Quotation
SOH	start of heading
STX	start of text
TCAM	Telecommunications Access Method
TCU	Transmission Control Unit
TP	teleprocessing
TWX	Teletypewriter Exchange
USASCII	United States of America Standard Code for Information Interchange

Chapter 1: Introduction

The IBM 3705 Communications Controller can be programmed to communicate with a large variety of remote computers, terminals, and transmission control units. This programming can be adapted to many different teleprocessing applications and operational requirements.

The network control program, as this programming is called, is first defined in the form of source statements, then generated by a compilation process, and finally loaded into the communications controller. These three steps—defining, generating, and loading—are the subject of this publication.

The Network Control Program

The network control program controls the transmission of data between the *host processor* and the remote stations in the teleprocessing network. (The host processor is the central processing unit to which the IBM 3705 is attached.)

The network control program recognizes and fulfills requests by the teleprocessing access method for data transfer to and from remote stations. In so doing, the network control program performs station polling, addressing, dialing, and answering, as appropriate for the type of station. Then it receives message data into buffers, inserts and deletes transmission control characters, and translates message data from processing code to transmission code (or vice versa). Finally, the network control program transmits the message data out of the buffers. These are the routine communication control functions that must be executed regardless of the kind of application fulfilled by the *teleprocessing subsystem*. (The teleprocessing subsystem consists of stations, data sets (or modems), communication lines, and the communications controller.)

The network control program also (1) performs automatic error recovery and statistical recording, (2) diagnoses controller, line and station malfunctions, and (3) changes operating parameters during program execution.

Communication Between Controller and Host Processor

Generally, communication between the network control program and the access method consists of an exchange of requests issued by the access method and responses returned by the network control program. Each request and response is accompanied by control information.

Defining the Network Control Program

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

Despite the numerous variables and options, a network control program is relatively easy to define. This is due to the modular design of the network control

program and to a high-level language used to specify the network configuration, operating parameters, optional message processing, and auxiliary functions.

A network control program is defined in the form of a source program consisting entirely of macro instructions called *network control program generation macro instructions*. These include configuration macros for specifying the elements of the teleprocessing network and are similar to those used in teleprocessing access methods. The source program, when punched into cards and preceded by the appropriate job control language statements, forms the input to the next step of the three-step process, the generation procedure.

Chapter 2, *Defining the Network Control Program*, describes each of the characteristics of the teleprocessing subsystem that must be considered in defining a network control program. Some of these are general characteristics common to most teleprocessing subsystems.

Other characteristics pertain to the capabilities and control techniques distinctive to the IBM 3705 and its network control program.

Once the characteristics explained in Chapter 2 are understood, the task of defining a network control program may be undertaken almost entirely by referring to the third chapter, *Network Control Program Generation Macro Instructions*.

Generating the Network Control Program

After the network control program is defined in the form of a source program containing network control program generation macro statements, it is ready to be generated. This is a two-stage procedure involving assembly and link-editing steps, performed under the System/360 Operating System (or its extension for System/370). The procedure may be executed in the host processor or in any other central processing unit that (1) can fulfill the operating system assembly and linkage-editing requirements and (2) has access to the IBM-supplied network control program module library. This library, supplied by the IBM Program Information Department, must be added to the operating system macro libraries before the generation process.

The primary output of the two stages of the generation procedure is a network control program load module, ready for loading into the communications controller. Chapter 4 describes the generation procedure.

Loading the Network Control Program

The final step in achieving an operating network control program is to load the network control program load module from the host processor into the communications controller. This requires that a loader utility program be executed in the host processor, with the controller online to the host processor. The network control program load module is loaded after a diagnostic routine verifies that the controller hardware is functioning properly. Loading may be done independently of the access method, using the loader utility included in the network control program support package; or, the access method may include a loader routine. Chapter 5 describes the loading procedure using the independent utility.

Obtaining the Contents of Controller Storage

A utility called the Dump program allows the contents of the controller storage to be transferred from the controller to the host processor, which then prints the contents in hexadecimal format.

The Dump program has two modules, one within the controller and the other within the host processor.

The host processor module transfers the controller module to the controller before the dump process begins. The two modules then work together to transfer the entire contents of controller storage to the host processor. Then the host processor module formats and prints as much of the storage contents as is desired. The teleprocessing access method is not involved in the dumping process.

After the contents of the controller have been passed to the host processor, the network control program must be reloaded into the controller. Teleprocessing operations can then resume.

Chapter 6 describes the operation of the Dump program.

Chapter 2: Defining the Network Control Program

Note: *The information in this chapter is preliminary and is to be used for planning purposes only. Any portion of the content is subject to change.*

This chapter describes the characteristics of the teleprocessing subsystem that must be considered in defining a network control program. The chapter is divided into five categories of characteristics that pertain to:

- The stations and lines of the teleprocessing network
- The communications controller
- Data transfer between the communications controller and the host processor
- Message traffic between the communications controller and the lines and stations of the teleprocessing network
- Message processing within the communications controller

The description of each characteristic is not exhaustive; it is intended to provide enough information to enable you to specify the appropriate parameters when coding the macro instructions given in Chapter 3.

For many characteristics, especially those relating to the equipment configuration, the decisions about what to code have been made by the *system designer*. (This is the individual who determines the teleprocessing equipment, network configuration, and communication services that constitute your teleprocessing subsystem.) You need only determine what these characteristics are and code the appropriate macros and operands accordingly.

Other characteristics relate to resources, such as the size of the buffers in the buffer pool, or to procedural options, such as the number of message blocks to be queued from a station before the network control program forwards them to the host processor. Such characteristics affect the message-handling capacity, throughput rate, and performance of the teleprocessing subsystem.

Once you are familiar with those characteristics that apply to your equipment configuration and application, you are ready to decide what to code in the network control program generation input deck. At this point you should go on to Chapter 3, which provides detailed information on how to code the macro instructions.

Teleprocessing Network Characteristics

The descriptions of the following characteristics give the names of the applicable macro instructions and operands. In many cases, particularly where lines and stations are involved, the characteristic can be specified in a macro instruction different from the one mentioned. (See *Teleprocessing Network Configuration* in Chapter 3.)

Station Characteristics

In this book, *station* refers to any equipment, regardless of type, that can transmit data onto, or receive data from, a communication line connected to the communi-

cations controller. This definition includes (1) computers, (2) transmission control units such as the IBM 2701 and 2703, (3) other 3705 controllers, and (4) the input/output devices (keyboards, printers, tape and card readers, punches, and display screens) usually referred to as *terminals*.

Each station that communicates with the IBM 3705 is represented by a TERMINAL macro instruction. Usually, a TERMINAL macro represents a single station. An exception is when stations of the same type, or of a limited number of types, communicate with the communications controller over the switched telephone network. A single TERMINAL macro can sometimes be used to represent all of these stations.

Type of Station

Type of station means the numerical designation by which it is known (for example, 1050, 2780, 3705). Appendix A gives a complete list of the types of stations with which the communications controller can communicate. Type of station is specified in the TERM operand of the TERMINAL macro, or, for certain types of stations, in the CUTYPE operand of the CLUSTER macro.

Terminal Features

For some types of terminals, the presence or absence of certain features of the terminal must be known to the network control program. The presence of any of the features listed below is specified in the FEATURE operand of the corresponding TERMINAL macro.

(The suboperand that specifies the presence or absence of the feature is given in parentheses at the end of each description.)

Transmit Interrupt (IBM 1050, 2741): If this feature is present on the terminal, the communications controller can interrupt a transmission from the terminal by sending the break signal. (BREAK or NOBREAK)

Buffered Received (2740 Model 2, 2792 Model 8, 3270): If this feature is present on the terminal, the network control program allows a time interval to elapse between successive transmissions to the terminal. During the interval, the network control program can communicate with other terminals on the same line. The interval is specified by the BFRDLAY operand of the TERMINAL macro. Operations with buffered terminals are explained under *Teleprocessing Subsystem Operation* in this chapter.

Conversational Mode (IBM 1050, 2740 Models 1 and 2 with Checking feature, 2770, and 2792 without Batch Message Input feature): A station equipped with this feature can receive message data, instead of the usual positive acknowledgment, in response to a message block the station has sent. The message block serves as the positive acknowledgment. Exchanging message blocks in this way improves line utilization because the time normally spent in re-addressing (re-selecting) the station is eliminated.

If this feature is specified, the network control program, upon receiving a message block from a station, sends the station a message block in response. If the pro-

gram has no data to send, it sends the positive acknowlegment instead. (CONV or NOCONV)

Dual Communications Interface (IBM 2780): If a terminal has this feature, the communications controller can use the Alternate Path Retry option (a procedural option). That is, if the nonswitched (leased) line over which the communications controller normally communicates with the terminal should fail, the communications controller can automatically attempt to reach that terminal over the switched telephone network. The network control program must be given the telephone number of the terminal, just as for any terminal that is connected to the switched network. (See the description of this below, under *Switched Point-to-Point Line*.) (DUAL or NODUAL)

Accelerated Carrier Return (IBM 1050): If your teleprocessing network includes IBM 1050 terminals having the Accelerated Carrier Return feature, you should specify this in the FEATURE operand of the TERMINAL macro for each terminal so equipped. The communications controller then sends a fewer number of idle characters than if the terminal did not have the feature, thus saving a small amount of transmission time whenever the new line character occurs in message text. (ACR or NOACR)

Checking, Station Control, Transmit Control (IBM 2740): The command sequence by which the network control program communicates with the IBM 2740 differs for each of these features or combinations of features. (CHECK or NOCHECK, SCTL or NOSCTL, XCTL or NOXTL)

Interrupt (IBM 2741), Receive Interrupt (IBM) 1050): If the terminal has this feature, it can interrupt the network control program while the program is sending to the terminal. (ATTN or NOATTN)

End of Transmission Character (IBM 1050)

You may specify that the EOB character, instead of the EOT character, is to signify end of transmission for messages from a 1050 terminal. Then the terminal operator need only enter EOB at the end of each transmission. To specify this option (it applies only to the IBM 1050), code ENDTRNS=EOB in the TERMINAL macro instruction.

Terminal Timeouts

The communications controller is provided with pre-set timeouts. When these expire, the network control program performs some action, such as breaking off contact with a station. This prevents a communication line from being idled indefinitely because of a station power failure or other malfunction. The network control program establishes a value of three seconds for the reply timeout, and 23.5 seconds for the text timeout, unless you specify a different value. To specify different timeout values, code the desired values in the REPLYTO or TEXTTO operands of the GROUP macro for the terminals for which the different values are to be used. All timeouts for terminals in the same line group must be the same.

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).

Line characteristics refer to (1) the physical and electrical attributes of the transmission path, such as its rated speed; (2) logical characteristics, such as the transmission code and line control scheme employed; and (3) related aspects of the line, such as the address by which it is known to the network control program.

Stations may communicate with the communications controller over any of three kinds of lines: nonswitched point-to-point, nonswitched multipoint, and switched point-to-point. You must code a LINE macro for each line connected to the communications controller, regardless of the type. This macro specifies to the network control program some (but not all) of the characteristics of the line.

Nonswitched Multipoint Line

For each multipoint line, the network control program must contain a service order table. This service order table has one or more entries representing each station and each separately pollable or addressable component on that line. The service order table determines the order in which the network control program attempts to establish communication with stations on that line.

For each multipoint line in the network, you must code, directly following the LINE macro, a SERVICE macro that defines a service order table.

Nonswitched Point-to-Point Line

To designate the communications controller as the secondary station on a nonswitched point-to-point line, code YIELD=YES in the LINE macro (or omit the operand). To designate it as the primary station, code YIELD=NO.

Whichever choice you make, the station at the other end of the line must be prepared to assume the complementary role (i.e., primary or secondary).

Except for the YIELD operand, you need code no other operands to designate the type of line as nonswitched point-to-point. The line is assumed to be of this type unless you explicitly code operands that specify another type.

For each nonswitched point-to-point line to which an IBM 1050 is connected, you must specify, following the LINE macro, a SERVICE macro that specifies the order in which the network control program is to service the components of the terminal.

Switched Point-to-Point Line

For each telephone line over which the communications controller may call stations, or receive calls from stations, you must code a LINE macro. In the GROUP macro that precedes the LINE macros for the switched lines, code

DIAL=YES. (The GROUP macro, rather than the individual LINE macros, indicates that the lines are switched. If any lines within the group are switched, all must be.)

You code a TERMINAL macro for each of the stations associated with the group. However, unlike nonswitched lines, no fixed relationship exists between a particular switched line and the stations with which the communications controller may communicate over that line. Stations are able to call the communications controller over any of the switched lines connected to it, and vice versa. You may, however, wish to limit certain lines to incoming calls, and other lines to outgoing calls. You may do so with the CALL operand of the LINE macro. Further, you may specify groups of lines (called dial sets) and associate them with specific stations. The network control program is then restricted to using a line in the specified dial set to call a station. The DIALSET macro specifies dial sets.

Half-Duplex vs. Full-Duplex Lines

The network control program must know whether a communication line is half-duplex or full-duplex. This is specified in the DUPLEX operand of the LINE macro. This information pertains to characteristics of the communication line, and not to the mode of transmission between the communications controller and stations. All stations with which the communications controller can communicate operate in half-duplex mode.

Line Speeds and Clocking

In the SPEED operand of each LINE macro, specify the speed at which the communication line is to operate. Determine this value from the system designer.

If the modem that connects the line to the controller has two possible data rates, you 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 or external clocking is used on the line. Internal clocking is provided by the communications controller; external clocking is provided by the modem.

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

Transmission Codes

The transmission code to be used for each station must be specified in the network control program generation input deck. Then the network control program can translate outgoing data characters from processing code (EBCDIC) to transmission code, and vice versa, for incoming data characters.

The transmission code used on a multipoint line must be the same for all stations on that line. You specify the code in the CODE operand of the corresponding LINE macro. (For BSC stations, the code you specify in that operand also informs the network control program which line control scheme is to be used. This is because the transmission code and the line control scheme are related.)

The transmission code used on a nonswitched or switched point-to-point line also is specified in the CODE operand, unless, for a switched line, Multiple Terminal Access is used. This facility is explained under *Network Control Program Procedural Options* in this chapter.

Line Address

Each communication line attached to the communications controller is identified to the network control program with a line address. You must specify this address in the ADDRESS operand of the corresponding LINE macro.

Communications Controller Characteristics

Several characteristics that must be identified to the network control program reflect the system designer's choice of hardware options for the communications controller. These are (1) the size of storage, (2) the type and number of channels that join the communications controller to the host processor, and (3) the type, number, and oscillator rates of the communication scanners installed. This information may be learned from the system designer.

The size of storage in K (1024) bytes must be coded in the MEMSIZE operand of the BUILD macro instruction.

In the CHANTYP operand of the BUILD macro, specify the type of channel adapter (1 or 2) with which the communications controller is equipped. If there is also a secondary channel, code SECCHAN=YES in the BUILD macro. (You need not specify the type for a secondary channel; it can only be a Type 2 channel.)

The communications controller can be equipped with from one to four communication scanners. The IBM 3705 Models A1 and A2 always have one scanner. Models B1-B4 can have up to two scanners; Models C1-C6 can have up to three scanners; Models D1-D8 can have up to four scanners. Each performs character service for up to four different speeds of lines; each speed requires its own oscillator within the scanner. For each scanner, the network control program must be told (1) the type of scanner, (2) the number of scanners, and (3) which oscillators each is equipped with.

This information, like the storage size and channel information, should be obtained from the system designer before you code the network control program generation macro instructions. You code the details of each scanner in the TYPE, MOD, and SPEED operands of a CSB macro—one macro for each scanner in the communications controller.

Interface Characteristics

This section explains those aspects of communication between the host processor and the communications controller that must be defined in the network control program.

Data Transfer from Host Processor to Controller

The host processor sends sequences of data to the communications controller at intermittent intervals. During one sequence of data transfer, the teleprocessing access method may send a request or a series of requests.

Although the network control program can receive data as long as the buffer supply lasts, the network control program should know how many buffers to allocate for each data transfer. Once these buffers are allocated, data transfer from the access method can proceed without further attention by the network control program supervisory routine.

The INBFRS operand of the HOST macro specifies the number of buffers to be initially allocated. Two factors should be considered when estimating a value for INBFRS.

If the size of a data transfer consistently exceeds the allocated buffer space, the network control program supervisory routine is frequently interrupted to provide more buffers for the excess data. As the proportion of time the network control program spends in allocating buffers increases, the network control program performance in general suffers.

On the other hand, if the amount of data received is consistently less than the allocated buffer space, a number of the buffers allocated is unneeded. Although the unneeded buffers are eventually used for receiving the next data transfer, their absence from the buffer pool reduces the number of buffers available for network operations. Buffer utilization is therefore less efficient.

In choosing a value for INBFRS, then, strike a reasonable balance between degraded network control program performance due to excessive demands on the supervisory routine, and unnecessary over-allocation of buffers.

Data Transfer from Controller to Host Processor

The communications controller sends a sequence of data to the host processor at intermittent intervals. During one sequence of data transfer, the controller may send a response or a series of responses.

In preparing to send to the host processor, the network control program must know how much data the teleprocessing access method can accept in one continuous transmission. The amount of access method buffer space available for receiving a data transfer must be specified in the MAXBFRU and UNITSZ operands of the HOST macro. MAXBFRU specifies the number of buffer units the access method allocates, and UNITSZ gives the size of each unit, in bytes. The total access method buffer space available is the product of the two values. (A buffer unit, in the access method, is the smallest amount of contiguous storage handled as buffer space; a buffer may consist of one or more units.)

In sending a series of responses to the host processor, the network control program causes the access method to begin receiving each successive response in a new buffer.

In some applications, the access method inserts prefixes in buffers ahead of the message data. A network control program option allows each new response block sent to the host processor to be offset, in the host processor buffer, by enough space to let the access method insert the desired prefix. The amount of offset is specified in the BFRPAD operand of the HOST macro. Two different values can be specified: one for the first buffer in which a request begins, and one for all subsequent buffers occupied by the request.

Network Control Program Procedural Options

Procedural options characterize the operation of the network control program once it has been defined. These options include (1) the manner in which the network control program handles startup and closedown of the subsystems, (2) the amount of data to be transmitted between stations and the communications controller at one time, (3) how much data is to be accumulated from a station before the network control program forwards it to the host processor, and (4) how many sessions are to be carried out concurrently on a multipoint line. By careful selection of these options, you can *customize* a network control program to best meet the requirements of your application.

Some network control program options require no more than a simple yes/no choice as to whether the option is to be included. Other options require you to choose from a range of values, such as the size of buffers, or the maximum amount of data to be transferred at one time between the communications controller and stations.

Some options require relatively little forethought before you decide what to specify. Other options require considerable attention to the performance implications of your choice.

Defining the Buffer Size

The network control program contains one buffer pool of fixed-size buffers. The size of the buffers (from 32 to 252 bytes, in multiples of 4 bytes) is specified in the BFRS operand of the BUILD macro. After loading the network control program, all storage space remaining after the network control program space requirements are met is devoted to the buffer pool. The space remaining, divided by the buffer size you have specified plus four bytes, yields the number of buffers in the pool.

Automatic Return of Text

For each request received from the host processor, the network control program may return a response indicating the result of the request. A request may be unsuccessful because (1) it is invalid, in which case the network control program rejects it outright, or (2) because an input/output error occurred during fulfillment of that request.

A request may be invalid for several reasons. The command code may be undefined; the request may specify a line or station not identified to the network control program during network control program generation; or the request is invalid for the type of station component it specifies (e.g., the request is a Write request issued for a terminal keyboard, or a Read request issued for a printer).

The network control program sends a response to the host processor indicating the reason for the non-fulfillment.

Unless you specify otherwise, the network control program does not return the text accompanying the request to the host processor when it returns the unfulfilled request. Instead, it releases to the buffer pool all buffers containing the text that accompanied that request.

If you wish the network control program to return the text along with the response, code the TEXTRET operand of the HOST macro. The network control program either returns all blocks, or only those blocks unsuccessfully transmitted, at your option.

Defining Interrupt Priority for Lines

The network control program is interrupted each time a data character is to be sent to, or received from, a communication line. If two or more interrupts occur together, the network control program must know the order, or priority, in which to process the interrupts. Therefore, an interrupt priority is associated with each line, via the INTPRI operand of the LINE macro. For a Type 1 communication scanner, the priority may be 0 or 1 (1 is the higher priority). For a Type 2 communication scanner, the priority may be 0, 1, 2, or 3 (3 is the highest priority). Since interrupts from higher speed lines require processing more promptly than those from lower speed lines, the priority specified should correspond to the line speed.

Logical Line Groups

You may wish to use the dynamic control facility to activate or deactivate, as a group, lines that are in different physical line groups (as defined by GROUP macros). If you plan to do so, define a *logical line group* by coding a LINELIST macro that specifies each of the lines you wish to be in that logical group. A line may be in more than one logical line group, although it cannot be in more than one physical line group.

This option can be of value, for instance, where lines to a particular time zone of the country are to be activated or deactivated together, and lines to the adjacent time zone need to be activated or deactivated an hour earlier or later. By using this option, you avoid having to individually activate and deactivate several lines.

Teleprocessing Subsystem Operation

This section describes the procedural options relating to operation of the teleprocessing subsystem (except for startup and closedown which are discussed in the two following sections).

Many of the procedural options in this section may be specified without regard to the type of line involved (that is, nonswitched point-to-point, nonswitched multipoint, and switched point-to-point). These options are described first.

Other options relate specifically to the three types of lines just mentioned. These are described following the general options.

Units of Data Transfer

Message data is transferred between the communications controller and communication lines in either of two logical entities for start-stop lines (blocks and messages), or three logical entities for binary synchronous communication (BSC) lines (blocks, messages, and transmissions).

A *block* is the smallest entity that the communications controller transfers to or from the communication line. It is defined as a sequence of text characters followed by an end-of-block character. This character is represented by EOB, for start-stop transmission, and by ETB, for BSC transmission. In this book, the abbreviation *EOB* is used to represent either EOB or ETB except where the context is specifically limited to BSC transmission. When the network control program is sending to a station, the program transmits the message text contained in the request, then an EOB or ETB character. When receiving from a station, the program accepts message text characters until it receives an EOB or ETB character. (See Figure 1 (a).)

A BSC message is a sequence of text, with or without intervening ETB characters, followed by an end-of-text (ETX) character. (See Figure 1(b).) When the network control program is sending to a BSC station, the program transmits the message text contained in the request, then transmits an ETX character. When receiving from a BSC station, the program accepts message text, including any intervening ETB characters, until it receives an ETX character.

A start-stop message is a sequence of text, with or without intervening EOB characters, followed by an end of transmission (EOT) character. When sending to a start-stop station, the network control program transmits the message text contained in the request, then transmits an EOB EOT sequence. When receiving from a start-stop station, the program accepts message text, including any intervening EOB characters, until it receives an EOT character.

A start-stop transmission is the same as a start-stop message.

A BSC transmission consists of a sequence of text characters, with or without intervening ETB and ETX characters, followed by an EOT character. When sending to a BSC station, the network control program transmits the message text contained in the request, then an ETX EOT sequence. When receiving from a BSC station, the network control program accepts message text characters, including any intervening ETB and ETX characters, until it receives an EOT character. (See Figure 1 (c).)

One of the important procedural options the network control program offers is the choice of unit of data transfer. That is, you may specify whether one block, one physical message, or one transmission is to be sent or received when the network control program receives a request from the host processor. This may be done separately for each station and component in the teleprocessing network.

When sending message data to a station, the request from the host processor specifies the unit of data transfer to be used.

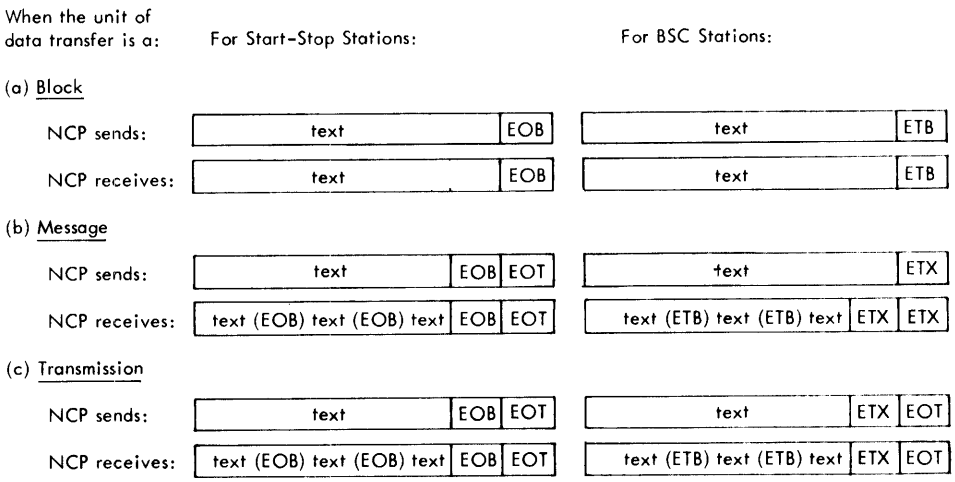


Figure 1. Units of Data Transfer

When receiving message data from a station, the request from the host processor *may* specify the unit of data transfer. If not, control information within the network control program specifies the unit to be used. You may specify, in the RECORD operand of the CLUSTER, TERMINAL and COMP macros, the unit of data transfer the network control program is to assume for any request that does not specify the unit. The possible values are BLOCK, MESSAGE, and XMISSION (transmission). If you do not specify the unit of data transfer, the assumed unit is a block.

When sending blocks, messages, or transmissions to a station, the network control program automatically sends the appropriate ending characters following the text, in accordance with the unit of data transfer specified by the request.

Preventing A Monopoly of Network Control Program Resources

The network control program provides two options that allow you to prevent one or several stations from monopolizing network control program resources to the exclusion of other stations. A monopoly could occur, for example, if a single station were permitted to send data to the communications controller indefinitely. These options limit the amount of processing time and buffer space the network control program devotes to a single station before it services other stations.

The network control program fills all requests for buffers from a single buffer pool and no station should monopolize the supply to the extent that other stations are excluded. Such excessive buffer usage could occur if the network control program were to accumulate too much data from a station before forwarding it to the host processor.

You can prevent buffer monopolization with two options. The first prevents excessive accumulation within the network control program of message data *from* a station. Normally, the network control program routine that receives data from a station accumulates an entire block of a message before sending that data to

the host processor. This is desirable because message processing routines within the host processor can then examine an entire block at once.

However, the network control program might receive a long message block, one requiring an unusually large number of buffers. To prevent this from happening, the network control program does not wait for an entire block to accumulate within network control program buffers. Instead, after the network control program has filled a certain number of buffers with incoming data, it sends the data in those buffers to the host processor.

Unless you specify a different value, the network control program will fill no more than seven buffers before passing them to the host processor. Meanwhile, it allows seven other buffers to be filled. If more than these seven additional buffers are needed to accommodate the block, the network control program once again passes them to the host processor, and fills seven more buffers. This sequence continues until the block ends, as indicated by an end-of-block character.

Each partial block passed to the host processor is called a *sub-block*. The response header that precedes each partial block indicates to the host processor that the data that follows is a sub-block, not a complete block.

You may specify a different number of buffers for each sub-block by coding the desired value in the TRANSFR operand of the LINE macro. This macro represents the line over which the station communicates with the communications controller. (The number you specify applies to all stations connected to that line.)

Limiting the number of buffers in this manner usually prevents undue buffer usage by any one station. This assumes, however, that the network control program promptly transfers the contents of the filled buffers to the host processor and then returns the buffers to the buffer pool. If for any reason the network control program cannot quickly transfer the partial blocks to the host processor (as when the host processor is slow to respond to the network control program's signal for service), the data accumulating from the station can still cause buffer monopolization.

For this reason a second option is available to limit the number of partial blocks received from the station. If this limit is reached, the network control program breaks off reception of data from that station. To limit the number of partial blocks, specify the desired value in the CUTOFF operand of the LINE macro instruction.

If you do not specify a limit, the network control program continues to receive from the station until an entire block has been received.

Programming Note: *The host processor program may avoid processing any of the sub-block data until the entire block is received. If any sub-block contains an error (indicated by a bit in the response header), the host processor program should be prepared to discard all of the sub-blocks in the block, not only the one in error.*

Buffered Terminals

Some types of IBM terminals receive incoming data into buffers at high speed, then print (or otherwise display) the data at a much slower rate. If the network control program sends successive data blocks to the same terminal, the line is idle for the entire time the terminal is printing. This is because the terminal cannot accept more data until the previous contents of the receiving buffer have been printed.

If the line is a multipoint line, the network control program can use the idle time to communicate with other terminals. At any given moment, the program can be sending to one terminal while other terminals are printing data received earlier.

To specify that the network control program overlap transmissions in this manner, code, in the BFRDLAY operand of the TERMINAL macro, the delay in seconds between successive transmissions to the same terminal. When choosing a value, consider (1) the number of terminals on the line, (2) the number that the network control program usually communicates with at a given moment, (3) the speed of the printer (or other output device), and (4) the average size of the message blocks.

The network control program automatically allows to elapse, between transmissions to the same terminal, the number of seconds you specify. The network control program performs this function when sending to an IBM 2770, 2980, 3284, 3286 or 2740 model 2.

Substitute Characters

The network control program automatically translates each character received from a communication line from its transmission code bit pattern into the EBCDIC equivalent. If the network control program detects an error in message text, it sets the *data check* bit in the response header for that block. If it can identify which character is erroneous, the network control program replaces that character with a substitute character. This may be either the EBCDIC SUB (substitute) character (hexadecimal 3F) or any other EBCDIC character of your choice. To specify a different character, code its hexadecimal EBCDIC equivalent in the SUBCHAR operand of the GROUP macro. The network control program uses the same substitute character to replace erroneous characters received from any station connected to lines in the line group.

Replacing invalid characters with a known substitute can be of value when processing the message block in the host processor. The processing program can scan every block containing an error (as indicated by the data check bit in the response header) for substitute characters. It then knows that each position containing the substitute character is in error and can react accordingly. The processing program may, for example, choose to ignore errors occurring in certain character positions.

The network control program can insert substitute characters only in message text received from those communication lines for which it performs parity (vertical redundancy) checking. This is true for most start-stop lines. Other kinds of

checking can detect the existence of an error, but can neither identify the particular character in error nor perform character substitution.

Sending WACK Sequences to BSC Stations

When receiving message text from a BSC station, the network control program may need to temporarily defer further input from that station. This can happen, for example, when the network control program does not have enough buffers to receive additional message text. When this occurs, the network control program responds to the block currently being received with a WACK sequence (wait-acknowledgment) instead of the usual positive acknowledgment (ACK-0 or ACK-1). The WACK sequence informs the sending station that the network control program is deferring the positive acknowledgment until it is again able to receive from that station.

When the station sends a WACK, the network control program responds with an ENQ character. If the station is still not ready to receive, it sends another WACK sequence, and the network control program again responds with ENQ; this exchange continues until the station is once again ready to receive. Then it sends the deferred positive acknowledgment (ACK-0 or ACK-1). To avoid prolonged holding of the line without text transmission, the network control program has a limit on the number of WACK sequences it will receive. Unless you specify a different value in the WACKCNT operand of the GROUP macro, the network control program will accept a maximum of 15 WACK sequences. If this limit is reached, the network control program will break the connection to the station. (Use of WACK sequences to defer message reception is possible only for BSC stations; this function is not available for start-stop stations.)

Sending Temporary Text Delay Sequences to BSC Stations

When the network control program temporarily suspends *sending* to a station, the network control program need not disconnect from that station. Instead it can send a temporary text delay (TTD) sequence in lieu of the next message block. The TTD sequence informs the receiving station that the network control program will continue sending after a short delay. The station responds to a TTD sequence with an ENQ character. Exchange of TTD and ENQ can continue for as long as the network control program needs to defer transmission or until the network control program has sent a maximum number of TTD sequences. As for WACK sequences, a limit is imposed to prevent long intervals when no message data is transferred over the line. Unless you specify a different value in the TTDCNT operand of the GROUP macro, the network control program sends a maximum of 15 TTD sequences. Thereafter it will disconnect from the station.

Error Recovery Attempts

When the network control program detects an error in message data received from a station, it automatically returns a negative response to the station. The station then re-transmits the block in error; the network control program re-executes the Read operation to receive the repeated block. Similarly, if the network control program receives a negative response from a station, it re-executes the Write operation to resend the block in which the receiving station detected an error.

The network control program may make several attempts to recover from the error by resending or re-receiving the block. After it has tried to recover a specified number of times, the network control program pauses for a specified interval, then makes more recovery attempts. Network control program generation parameters tell the network control program (1) how many successive attempts to make, (2) how long to pause before beginning a new sequence of attempts, and (3) the total number of sequences of attempts to execute. The last two apply to WRITE operations only.

Assume the number of attempts per sequence is three, the pause is seven seconds, and the total number of sequences is two. The network control program, upon detecting an error, makes up to three attempts to recover, pauses seven seconds, then makes up to three more attempts. If these attempts are unsuccessful, the network control program pauses again for seven seconds, then makes up to three final attempts. (The number of sequences specified—in this example, two—does not include the initial recovery sequence.)

If all attempts fail, the network control program ends the Read or Write operation and sets the data check flag in the response header for the corresponding request. In the preceding example, the network control program would set the data check flag after ten attempts to recover—the original transmission plus nine retransmissions.

You may specify values for the above three parameters either in the network control program generation input deck (via the RETRIES operands of LINE macros), or during execution of the network control program with the dynamic control facility. To permit dynamic changing of the limit for recovery attempts, include RETCNT among the options you specify in the SYSCNTRL macro.

Nonswitched Point-to-Point Lines

Some BSC stations can be equipped with a dual communications interface feature. This permits the stations to communicate with the communications controller over an alternate switched-line connection if the primary (nonswitched point-to-point) line should fail.

To accommodate this feature, the network control program has an option called *alternate path retry*. Using this option, the network control program automatically establishes contact with the station over an alternate switched line if it cannot communicate successfully over the primary line. The network control program calls the station over the alternate line when error recovery attempts for the primary line are exhausted; that is, at the moment when the network control program would notify the host processor if there were no alternate line.

The network control program also calls the station over the alternate path when the line fails after some message text has already been transmitted.

In this case, if the network control program were *sending* to the station when the line failed, the first block the network control program sends after re-establishing communication is the one following the last successfully transmitted block.

Once the alternate connection is established, the network control program communicates with the station over that line until the end of the session, or until the alternate path also fails. The network control program attempts to use the primary line again for the next session. (A *session* is a series of command and data interchanges between the host processor and a teleprocessing device.)

To use the alternate path retry option, you must specify a dial set from which the network control program is to select a line as the alternate path. The lines in the dial set must have the same characteristics as the primary line. The DIALALT operand of the LINE macro for the primary line specifies the name of the DIALSET macro for the dial set.

You must also specify, in the TERMINAL macro for the station, the telephone number of that station.

Finally, you must include DUAL among the parameters you code in the FEATURE operand of the TERMINAL macro. This informs the network control program that the station has the dual communications interface feature, which is a prerequisite for use of the alternate path retry option.

Nonswitched Multipoint Lines

Regardless of whether there is one station on a communication line (as on a point-to-point line) or more than one station (as on a multipoint line) the network control program communicates with just one station at any given instant. That is, only one *session* can be active at any one moment. However, the network control program can communicate *concurrently* with many stations on that line. It does so by dividing the time it spends with any one station into segments, and interspersing these segments with time segments devoted to communication with other stations.

Each session is said to be *active* when the network control program and the station are actually communicating; a session is *suspended* during the time the network control program is engaged in an active session with another station. Several sessions may simultaneously be suspended but only one may be active at any moment.

Establishing *multiple* sessions is desirable for better line utilization. Transmission between the network control program and any given station is seldom continuous. This is especially true where the station receives data into a buffer at high speed, then prints it at a much slower speed. The line cannot be used to send more data to that station until the printing operation is completed. The line is therefore idle.

Idle line time often occurs even when the station is not equipped with a buffer but is used for inquiry and response operation. Upon receiving a response from the network control program, the terminal user pauses several seconds before entering another inquiry. The line is idle during this time.

When the network control program is allowed to establish multiple sessions, the line utilization increases because the network control program uses the line time (otherwise idle) to communicate with other stations on the same line.

The decision whether to have the network control program establish single sessions or multiple sessions on a multipoint line is based on many application-dependent factors. But in no case is the program in the host processor aware of, or affected by, the choice you make.

Any number of sessions may be established, up to the total number of components attached to stations on the line. However, establishing concurrent sessions with several components of the same station is usually inadvisable, because concurrent operation of the components may be confusing to the operator at the station.

In the SESSION operand of the LINE macro, you may specify a maximum number of sessions the network control program will concurrently maintain on that line. This is called the *session limit*. If you omit the SESSION operand, the network control program maintains only one session at a time on the line.

If the network control program currently has requests to establish sessions for two components on one station and for one component on each of three other stations, the number of sessions the network control program establishes (if able) is five, unless the session limit is set at some lower value.

If the network control program is unable to honor a request for a station because the session limit has already been reached, it places the request on the request queue for the station. The network control program establishes the requested session later, when completion of some other session causes the number of sessions to drop below the session limit.

During the time the network control program is not actively engaged in one of the active sessions, it attempts to establish a session for each of the remaining requests (unless the session limit is already reached). The interval during which it attempts to begin the new sessions is called *service seeking*.

Servicing existing sessions and service seeking both alternate in cycles. The network control program devotes the first part of each cycle to servicing the existing sessions. After each session has been serviced once, the network control program enters the second part of the cycle to perform service seeking if it has any outstanding requests to begin sessions, and the session limit has not been reached. If there are no requests, the network control program skips the second part of the cycle and begins servicing existing sessions again.

The *cycle* mentioned does not represent any particular time interval. The elapsed time between the beginning of successive cycles depends entirely upon (1) the number of sessions currently established, (2) the amount of data transmitted during each, and (3) the amount of time the network control program spends seeking to establish new sessions.

The last factor may be controlled via the SERVLIM operand of the LINE macro for the line. The session limit minus the number of current sessions is the number of sessions the 3705 may establish during service seeking. One service seeking attempt is made for each session that the 3705 may establish during service seeking. The SERVLIM operand of the LINE macro is the number of devices the

3705 will try to contact during each service seeking attempt. Therefore, the maximum number of devices the 3705 will try to contact in performing service seeking is the SERVLIM times the number of service seeking attempts.

If you expect that (1) a line will deliver little or no message data to the communications controller for extended periods, and (2) during the same period the network control program will not receive requests to send to any station on the line, you may minimize the amount of time the network control program spends in nonproductive service-seeking. To do so, specify that the network control program pause for several seconds or minutes between successive service-seeking operations. This pause is in effect only if no sessions currently exist on the line. Once the network control program has established a session (either because some station responded positively to polling or because the network control program received a request to contact a station), the service-seeking operation, if required, resumes immediately when the network control program finishes servicing the existing session.

To specify a pause, code in the PAUSE operand of the LINE macro a value equivalent to the number of seconds the network control program is to pause between successive service-seeking operations. The maximum pause you can specify is 255 seconds (four minutes 15 seconds). If you omit this operand, there is no pause.

The amount of data to be sent and/or received by a station during each active session is specified as a number of transmissions (that is, a sequence of message text followed by an EOT character). If you choose to allow multiple sessions on a multipoint line, you should specify a limit on the number of transmissions to be sent or received during each active session. The limit should be less than the total number of transmissions expected in a typical session. You may specify this value separately for each station and station component connected to the line via the XMITLIM operand of the corresponding TERMINAL and COMP macros. This limit governs transmissions in either direction between the communications controller and the station. A transmission limit of three, for example, has been reached when the network control program has sent three transmissions to the station, or received three transmissions from the station, or sent two and received one, etc.

Note: *If you specify, via the SESSION operand, that you want the network control program to establish multiple sessions on a line but do not specify a transmission limit, the network control program establishes only one session on that line. This is because the network control program observes no limit unless you specify one. If there is no limit, the session proceeds without interruption until all requests for that station have been fulfilled, unless the session has been suspended due to a Negative Response limit having been reached.*

Switched Point-to-Point Lines

The following sections discuss procedural options relating to switched point-to-point lines.

ID Exchange and Verification (BSC Stations Only)

The network control program can receive an identification (ID) sequence from any BSC station that calls the communications controller, and can check that sequence against a list of sequences within the network control program to see if it is valid. If the network control program does not recognize the sequence as valid, it does not proceed with message transmission. Instead the network control program waits for the calling station to break the line connection, or it maintains the connection but forwards the unrecognized sequence to the host processor for examination. (This is a choice you specify in the network control program generation input statements.) In the latter case, the access method can inspect the sequence and then inform the network control program whether to break the connection or to proceed with message transmission.

If the network control program does the checking, the elapsed time between receiving a call and proceeding with message transmission is shortened. But the list of valid sequences requires storage space in the network control program. The space needed can be considerable if the number of valid ID sequences in the list (that is, if the number of stations that may call the communications controller) is large.

A compromise is to have the network control program check the sequences for stations that call most often, and forward to the host processor sequences received from stations that call less frequently. The network control program checks a station's ID sequence only when the station calls the controller. The program does not check ID sequences when it calls a station.

The various ID verification options are specified with the IDSEQ operand of the LINE macro and the NOMATCH operand of the IDLIST macro.

Sending the Controller ID Sequence to Stations

Just as a station may send its ID sequence to the communications controller, you may specify a communications controller ID sequence that the network control program will send to BSC stations.

The maximum length of the ID sequence for the communications controller is 15 characters. However, different types of stations expect different length ID sequences. For each station expecting the controller ID, you must specify how many of the communications controller ID characters the network control program is to send.

To assign the communications controller its ID sequence, code the CUID operand of the BUILD macro. You may specify any combination of EBCDIC alphabetic and numeric characters up to the limit of fifteen.

For each BSC station to which the network control program is to send the communications controller ID sequence, code in the CUIDLEN operand of the TERMINAL macro the number of characters the network control program is to send. The network control program will then send that many characters, beginning with the high-order (leftmost) character.

Number of Attempts to Dial a Station

Unless you specify otherwise, the network control program, upon receiving a request to call a station, automatically dials the station's telephone number up to four times in succession. If the last attempt is unsuccessful, the network control program returns a response to the host processor indicating the fact. To increase or decrease the number of attempts to dial the station, code the desired value (up to 255) in the REDIAL operand of the LINE macro. (A value of 255 indicates that the network control program will redial the station indefinitely until the station answers or the teleprocessing access method resets the request.)

Dial Sets and Alternate Switched Line Operation

A *dial set* is a logical grouping of switched point-to-point lines, any one of which the network control program can use to call a station. By logically associating many stations with the same dial set, you can have relatively few lines serve a large number of stations. (See Figure 2.) This economy of lines is possible because the network control program communicates with any given station for only a small percentage of time. A dial set is defined by the DIALSET macro; in the LINES operand, you list the lines the dial set is to contain.

Every switched line that the network control program can use to *call* a station must belong to one, and only one, dial set. (That is, every line for which the LINE macro specifies CALL=OUT or CALL=INOUT must belong to a dial set.) Lines used *only* to receive incoming calls from stations (CALL=IN in the LINE macro) need not belong to a dial set.

Each TERMINAL macro for a station that may be dialed over the switched telephone network must specify the name of a dial set. When the network control

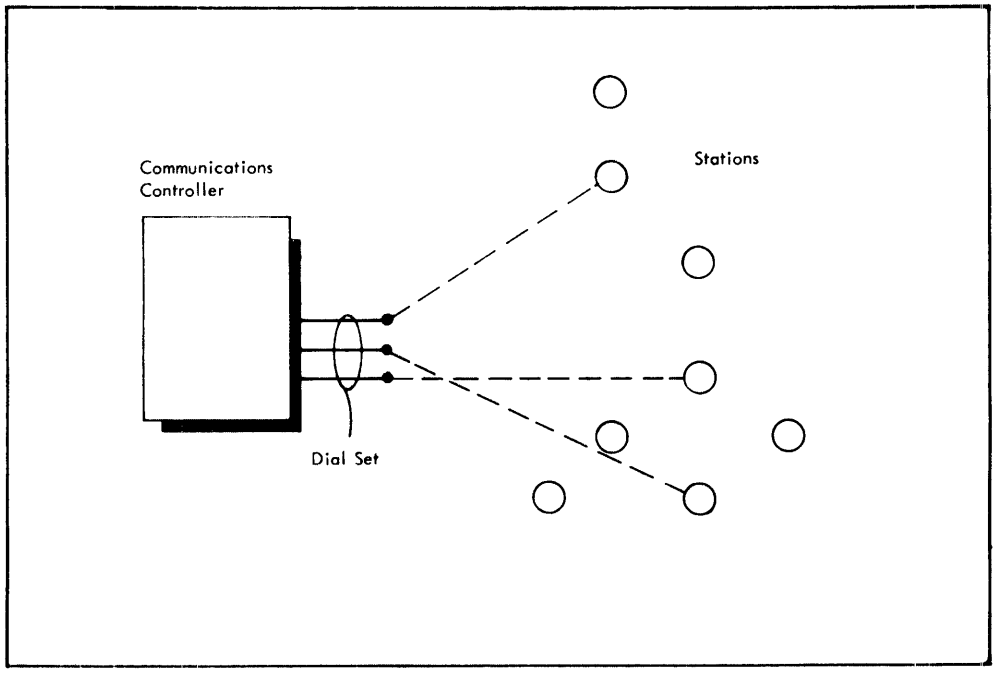


Figure 2. Using a Dial Set to Serve Many Stations

program receives a request to call a station (a *call-out* request), it calls that station over any available line in the specified dial set. (A line is available if it is not busy with another call, and it is not logically deactivated.)

If all lines in the dial set are busy (or all lines not busy are logically deactivated), the network control program cannot make the call. Instead, the program must wait until a line becomes available. The program places the call-out request on the request queue associated with the dial set. When a line in the dial set becomes available, the program uses it to fulfill the call-out request that has been longest on the queue. To avoid accumulation of an excessive backlog of unfulfilled requests, a *queue limit* is established. When this limit is reached, the network control program will accept no further call-out requests from the host processor. The network control program sends a response to the host processor indicating the reason for the rejection.

The queue limit is established by the QLIMIT operand of the DIALSET macro.

Reserving Lines for Incoming Calls: If some or all of the lines in a dial set are used for incoming calls as well as outgoing calls (CALL=INOUT in the *Line* macro), you may wish to reserve some percentage of the lines for incoming calls only. This prevents call-out requests from monopolizing all of the lines in the dial set so that stations are unable to call the communications controller.

The RESERVE operand of the DIALSET macro specifies the number of lines to be reserved for incoming calls. Note that the network control program does not reserve *specific* lines. It avoids using any of the remaining available lines to call stations if the number of lines available has decreased to the value specified in the RESERVE operand.

Alternate Dial Sets: The queue of unfulfilled *call-out* requests may become long as the network control program receives requests faster than it can complete calls in progress. When this happens, fulfillment of the queued requests may be considerably delayed.

To minimize such delays, you may specify an *alternate* dial set for each dial set.

For each call-out request it receives, the network control program looks for an available line in the first dial set. If it finds none, the program looks for an available line in the alternate dial set.

To associate two dial sets, code the name of one DIALSET macro in the DIALALT operand of the other DIALSET macro. The primary dial set is the one whose DIALSET macro specifies the name of the alternate DIALSET macro, thus:

```
PRI     DIALSET   LINES=(...),DIALALT=ALT
ALT     DIALSET   LINES=(...)
```

You may associate several dial sets in this way, such that each DIALSET macro except the last specifies the name of an alternate DIALSET macro:

```
DSET1   DIALSET   LINES=(...),DIALALT=DSET2
DSET2   DIALSET   LINES=(...),DIALALT=DSET3
DSET3   DIALSET   LINES=(...),DIALALT=DSET4
DSET4   DIALSET   LINES=(...)
```

The DIALSET macro for the alternate dial set must immediately follow the DIALSET that refers to the alternate dial set, as shown in the four DIALSET macros above.

The last DIALSET macro in the "chain" must not contain a DIALALT operand.

When you set up multiple dial sets, each is an alternate for all of the others preceding it in the chain. In the example, if dial set 1 (DSET1) has no available line when a call-out request is received, the network control program uses a line in dial set 2, if one is available. If no line is available, the network control program looks for an available line in dial set 3, and so on.

You may wish to allow a queue of call-out requests to develop for the primary dial set before the network control program attempts to use a line in the alternate dial set. The number of call-out requests the program will allow to accumulate is specified by the QLOAD operand of the DIALSET macro.

If for example you code QLOAD=3, the program will use the alternate dial set only if the number of call-out requests awaiting service by the primary dial set exceeds three. The request that has been longest on the queue is serviced first.

The value you specify in QLOAD must be less than the value you specify in QLIMIT. Otherwise, the network control program never accepts enough requests from the host processor to allow the alternate dial set to be used.

Teleprocessing Network Initiation

Once the network control program has been loaded, the host processor must send requests to activate some or all of the communication lines in the teleprocessing network before message transmission can begin.

The stations and components connected to the network must also be logically activated in order for the communications controller to communicate with them. A network control program option allows you to specify whether a station or component is to be logically activated when the network control program is loaded. Unless you specify that a station or component is to be initially inactive (by coding ISTATUS=INACTIVE in the TERMINAL or COMP macro), the network control program considers it to be logically active. The network control program will thereafter accept requests for the station, provided that the host processor has logically activated the line to which the station is connected.

No message transmission takes place until the host processor sends a request for the station or component, with one possible exception.

A network control program option is available that causes the network control program to issue an invite request, for multipoint lines only, instead of waiting for the host processor to specifically issue an invite request for that station.

If you wish the network control program to issue invite requests for a station or component, code ACTIVE and INVITE in the ISTATUS operand of the TERMINAL or COMP macro.

You may also specify the modifier for the invite request, just as an invite request from the host processor contains a modifier. These *modifiers* specify the data unit (block, message, transmission) to be received via the invite request, and the action the network control program is to perform after that unit has been received (disconnect the station, or disconnect the station and then reissue the invite request).

You specify the request modifier in the ISTATUS operand of the TERMINAL macro.

Restart Using Checkpoint Records

After loading the network control program into the communications controller, the host processor must send requests to the network control program to activate lines and stations before message transmission can begin (except for network control program-issued requests on multipoint lines).

One of the network control program options is *checkpoint/restart*. If the network control program periodically sends checkpoint records to the host processor indicating the current status of the lines, stations, and components in the network, the host processor can—after loading the network control program—re-establish the former status of the network. It does so by sending to the network control program restart requests that convey the latest checkpoint records.

The host processor may not have checkpoint records with which to re-establish the network status, or it may choose not to restart the network control program using the checkpoints. In either case, the host processor may initially establish the status via control requests, or may begin sending teleprocessing requests.

The network control program sends checkpoint records to the host processor only when it receives a control request that specifies a checkpoint record is to be sent. Checkpoint records are always sent as data following the response header that conveys the result of the request that specified checkpoint recording.

To specify that the network control program is to respond to requests for checkpoint recording, code CHKPT=YES in the BUILD macro. Otherwise, the network control program will ignore any requests for checkpoint records.

Teleprocessing Network Closedown

Message traffic in the teleprocessing network stops when one of the following occurs.

- The host processor logically deactivates all lines via one or more control requests, or when the host processor sends a *closedown* control request.
- Closedown is initiated from the communications controller operator panel.
- The network control program itself initiates closedown when it can no longer communicate with the host processor because of CPU or channel failure.

In the last two cases, the network control program may send a *critical situation* message to all stations for which a session is active. If the station is idle, the network control program sends it the message immediately. If the station is busy receiving or sending, the network control program concludes the current operation, then sends the message to that station.

If you wish the network control program to send *critical situation* messages to any or all stations, code the desired text of the message in the CSMSG operand of the BUILD macro. Then, for each station you wish to receive the message, code CRITSIT=YES in the corresponding TERMINAL macro. When automatic (network control program initiated) closedown occurs, the network control program will send the message to all active stations.

Regardless of what you code in the CSMSG operand, the critical situation message begins with the date and time (the time is in 24-hour format) and ends with the text you specify in the CSMSG operand.

Example: 05/14/72 19.27.05 NO FURTHER TRANSMISSIONS UNTIL NOTIFIED.

(The network control program does not automatically send a message to notify stations that transmission may resume; this is the responsibility of the host processor.)

Checkpoint Records

Once the teleprocessing network is closed down, certain network status information must be restored before message transmission can resume.

It may be desirable to have lines and stations assume the same status as they had at some moment before closedown. If so, specify CHKPT=YES in the BUILD macro. Then the host processor can accumulate checkpoint records periodically for any lines, stations, and components that may later be restored to the checkpointed status.

To restore the status, the host processor must send restart requests to the network control program immediately following reloading of the network control program. Each restart request includes one of the checkpoint records taken earlier.

As it receives checkpoint records for various resources, the network control program automatically restores the status of the resource to the checkpointed values.

As long as the network control program receives only restart requests from the host processor, it remains in *restart* mode. After the host processor has sent all the checkpoint records it wishes to send, it sends whatever control and teleprocessing requests are appropriate to resume message transmission. The first request the network control program receives that is not a restart request removes the network control program from *restart* mode. Thereafter, it rejects any subsequent restart requests.

Block Handling Options

Block handling refers to the optional processing of message data within the communications controller. The network control program can process either message data received from the host processor for retransmission to a station, or message data received from stations for forwarding to the host processor.

The IBM-supplied network control program modules provide two standard message processing functions. Each is performed by a block-handling routine invoked by a network control program generation macro instruction. In addition, user-coded block handling routines may be added to the network control program during the generation procedure. A network control program generation macro, UBHR, allows you to invoke the user block-handling routines in the same way as IBM-provided block-handling routines. The IBM-supplied block-handling routines cannot, however, be included in a user block-handling routine.

The optional processing functions the network control program can perform are as follows.

Insertion of Date and Time

The network control program can insert the current date, or time of day, or both, into message blocks it receives from a station or from the host processor. The date may be in either of two formats: month/day/year, for example, 10/18/72, or year followed by day of year, for example, 72.289. The time is in the format hh.mm.ss (hours, minutes, seconds). The continental (24-hour) form is used. For example, 09.17.25 and 21.17.25 represent 9:17:25 a.m. and 9:17:25 p.m., respectively. (Each format is preceded by an EBCDIC blank character.)

The date and time may be placed in the first block of each message or transmission, or in every block of the message or transmission.

Date and time insertion is specified with the DATETIME macro.

Automatic Text Correction

With this editing function, the network control program replaces message characters incorrectly entered from a terminal keyboard with the corrected characters the operator subsequently sends. The network control program does this by scanning each block for predefined characters called text canceling characters. The network control program deletes from the block each such character it finds, plus one preceding text character. For example, if it finds a sequence of three canceling characters, it deletes those three characters plus the three immediately preceding characters.

A keyboard operator may enter COMMUNCIATE and, seeing he has misspelled it, enter five backspace characters to *back up* to the first erroneous character. Then he re-enters the corrected characters, thus:

COMMUNCIATE bksp bksp bksp bksp bksp ICATE

If you have chosen the text correction *(editing)* option and specified *backspace* as the text canceling character, the network control program deletes the five backspace characters and C I A T E. The remaining letters form the correctly spelled word, COMMUNICATE.

The text canceling character need not be a backspace character. Any other character (except a line control character) is adequate if it is not used in any other way within text. For example if / is the character chosen, and a keyboard operator enters ATLANITC///TIC, the editing block-handling routine corrects the word to ATLANTIC.

The EDIT macro specifies the text correction function.

User Block Handling Routines

Any block handling routine you provide is referred to as a *user* block handling routine. You code a user block-handling routine using the IBM 3705 assembler language (similar to the Operating System assembler languages) and place the routine in a data set available to the network control program generation procedure. Then you include in the network control program generation input deck a UBHR macro instruction that specifies the routine and the point at which the network control program is to execute it.

Associating Block Handling Routines with Stations

The requirements of the teleprocessing access method determine how the network control program should process messages before sending them to the teleprocessing network or the host processor. The requirements may differ for different stations, or for different components of a station. You may, for example, wish to provide the text correction function for messages entered from a terminal keyboard, but not for messages received from a tape reader. Or, you may wish to insert time and date information in messages received from station 'A' but not in those received from station 'B'.

Network control program generation macro instructions provide a means of grouping individual block-handling routines into block handlers, and for combining block handlers into block handler sets. Block-handler sets can then be associated with individual stations or station components. Each block handler within a set can be executed at a different logical point in the flow of data through the communications controller.

For instance, one block handler in the set can be executed immediately upon arrival of a message from the host processor, before the network control program is ready to send the message to the station (i.e., before the network control program has obtained a communication line). Another block handler in the same

set can include block-handling routines that process message data from the host processor *after* the network control program has obtained a line.

The block handler may also include routines that process message data from a station *before* the network control program releases the line over which the data was received.

A third block handler in the set may be assigned to process message data received from a station after the network control program has made the line available for communicating with other stations.

The network control program generation macro instructions for grouping block handling routines into block handlers are STARTBH and ENDBH. A third macro, BHSET, combines block handlers into sets.

To assign block handler sets to stations or station components, you code the name of the set in the BHSET operand of the appropriate CLUSTER, TERMINAL, or COMP macro instruction. In the BHEXEC operands of the same macros, you specify which block handlers within the set are to be executed at the logical points in the message flow.

Diagnostic and Service Aids

The network control program diagnoses problems by means of two options: online terminal testing and address trace. They are useful in identifying malfunctions within the teleprocessing subsystem and network control program. Although optional, their inclusion in the network control program is recommended.

Online Terminal Testing

Online terminal testing is a diagnostic aid by which a terminal or console may request a variety of tests to be performed upon either the same terminal or console or a different one. The terminal requests the test by entering a test request message having a defined format. The requested test is performed, and the results are printed at the terminal undergoing the test, at the terminal requesting the test, or a different terminal altogether.

If the online terminal test option is specified, the network control program recognizes the test request message and passes it to the teleprocessing access method, just as it does a normal message. The network control program does not, however, perform any block processing upon the message, but sends it unchanged to the access method. The network control program identifies the message as a test request message for the host.

From this identification, the teleprocessing access method detects that the message requests the online test function, and interprets the parameters within the message to determine the kind of test to perform. The access method then selects the appropriate test modules and sends a series of interpretive commands to the network control program that indicate what teleprocessing operations to perform. The network control program executes the operations and returns responses as

necessary to the access method. The access method examines the responses and sends the network control program other interpretive commands, as necessary.

The network control program is only an intermediary in online test operations. It recognizes test request messages, routes them to the host processor, recognizes interpretive commands from the host processor, and executes teleprocessing operations accordingly. Online test operations require buffer space to hold the interpretive commands and an online test control block. The buffers are required only for the duration of the test operation.

The network control program can execute online test teleprocessing operations for any number of lines concurrently; some additional buffer space is needed for each additional line being tested. Teleprocessing operations on lines not undergoing testing are unaffected.

The OLT operand of the BUILD macro specifies whether the online test facility is to be included in the network control program.

Address Trace

Address trace is a service aid by which the contents of selected areas of communications controller storage and selected external registers can be recorded at each successive interrupt. Certain types of interrupts, or all interrupts can be designated. The network control program records the address trace data in a trace table within communications controller storage. When the desired data has been recorded, the contents of the trace table can be displayed on the control panel of the communications controller. The contents of communications controller storage can be transferred to the host processor via the Dump program and the contents of the trace table examined in the dump listing.

The TRACE operand of the BUILD macro specifies whether the address trace option is to be included in the network control program, and specifies the size of the trace table.

System Definition Macro Instructions

BUILD
SYSCNTRL

Configuration Definition Macro Instructions

HOST
CSB
IDLIST
LINELIST
SERVICE
LCST
MTALIST
MTAPOLL
DIALSET
TRANSTBL

Teleprocessing Network Configuration Macro Instructions

GROUP
LINE
CLUSTER
TERMINAL
COMP

Block Handler Definition Macro Instructions

STARTBH
ENDBH
DATETIME
EDIT
UBHR
BHSET

Generation Delimiter Macro Instruction (GENEND)

GENEND

Chapter 3: Network Control Program Generation Macro Instructions

Note: *The information in this chapter is preliminary and is to be used for planning purposes only. Any portion of the content is subject to change.*

This chapter gives detailed descriptions of the macro instructions with which you define the network control program.

Macro Instruction Coding Conventions

The following conventions are used in the descriptions of the network control program generation macro instructions.

- Capital letters represent values you code directly, without change.
- Small letters represent parameters for which you must supply a value.
- Brackets [and] enclose operands or symbols that are either *optional* or *conditional*.
  An optional operand is one that you may choose to code or to omit, independent of other operands you may code or omit. Depending on the operand, omitting it may cause network control program coding for the corresponding feature or function to be omitted or included, or omitting it may cause a specific numeric value to be assumed. The assumed value is always given.
  A conditional operand is one that you may need to code or to 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.
- Braces { and } 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 network control 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 (that is, BHSET=DYNAMIC specifies that the BHSET is dynamic; BHSET='DYNAMIC' specifies that the *name* of the BHSET is DYNAMIC.)
  Symbols coded in the name field of a macro instruction must not begin with a $ character.
  Within the macro instruction formats and descriptions, operands that are always required appear first, in alphabetical order. All remaining operands (conditional and optional) appear next, in alphabetical order.
  Data set names must begin with an alphabetic character or $, @, or #.

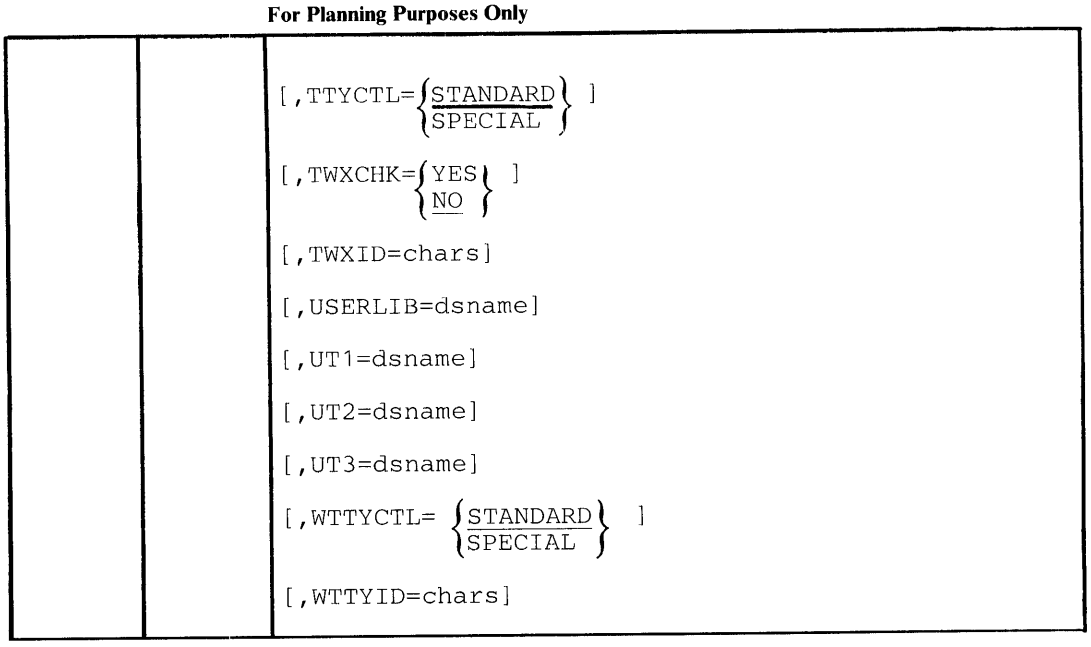
System Definition Macro Instructions

BUILD Macro Instruction

The BUILD macro specifies:

- The size of buffers in the buffer pool.
- The name that is to be assigned to the network control program load module.
- The type and number of channel adapters in the communications controller.
- The controller storage size.
- Certain optional facilities that may be included in the network control program.
- The names of data sets used in the generation process.

Name	Operation	Operands
[symbol]	BUILD	MEMSIZE=n, NCPLIB=dsname, OBJLIB=dsname [,ANS={YES | NO}] [,BFRS={size | 60}] [,CHANTYP=([ptype][,stype])] [,CHKPT={NO | YES}] [,CSMSG='text'] [,CUID=chars] [,NEWNAME={NCP001 | symbol}] [,OLT={YES | NO}] [,QUALIFY={symbol | NONE | SYS1}] [,RESET={NONE | ([IMM][,COND][,DEVQ][,OLT][,BLOCK]) | ALL}] [,SECCHAN={INACTIVE | BACKUP}] [,SLODOWN={12 | 25 | 50}] [,TRACE={NO | (YES[,{size | 10}])}]



`[symbol]`

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

`MEMSIZE=n`

Specifies the storage size, in K (1,024) bytes, of the controller.

Example: If the storage size is 48K, code MEMSIZE=48 (omit the K). Valid sizes are 48, 80, 112, 144, 176, 208, and 240 (K).

This operand is required.

`NCPLIB=dsname`

Specifies the name of a partitioned data set that will contain the network control program load module and the Resource Resolution Table produced by 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 character must be alphabetic.) This data set must be cataloged.

This operand is required.

`OBJLIB=dsname`

Specifies the name of a data set that will contain the output from all assemblies during Stage 2. (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.)

This operand is required.

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

Specifies whether or not Automatic Network Shutdown is to be included in network control program.

Code ANS=YES (or omit the operand) if you wish the facility to be included. Code ANS=NO if you do not wish it included.

```
[BFRS= {size}]
       {60 }
```

Specifies the size, in bytes, of buffers in the network control program buffer pool. The minimum size is 32 bytes, if the online terminal test facility is not included in the network control program (OLT=NO in this macro). If the online terminal test facility *is* included (OLT=YES or OLT operand is omitted), the minimum size is 36 bytes. The maximum size is 252 bytes. (The network control program generation procedure adds 4 bytes to the size specified. These bytes are used for control purposes.)

```
[CHANTYP=( [ptype][,stype] )]
```

Specifies how many channels are installed and the type of each.

ptype
: Specifies the type of the primary channel (i.e., the channel which will be used for IPL) as TYPE1 or TYPE2. If 'ptype' is omitted, the primary channel is assumed to be TYPE2.

stype
: Indicates that the secondary channel is installed and identifies the type as TYPE1 or TYPE2. If 'stype' is omitted, it is assumed that a secondary channel is not installed. If 'ptype' is specified as TYPE1, 'stype' may not be specified as TYPE1.

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

Specifies whether or not the checkpoint/restart facility is to be included in the network control program. Code CHKPT=YES to include the facility; code CHKPT=NO (or omit the operand) to omit the facility.

```
[CSMSG='text']
```

Specifies the text of the "critical situation" message to be sent to the active stations before automatic network shutdown occurs. The message will be sent to each station whose TERMINAL macro specifies CRITSIT=YES, if both the station and the line are logically activated. The maximum number of characters, including the framing quote marks, is 256.

If this operand is omitted, the network control program will not notify stations before automatic network shutdown occurs.

[CUID=chars]

Specifies the characters in the controller identification sequence that the network control program may send to BSC stations on switched lines. You may specify a maximum of 15 characters. (The network control program will send some or all of these characters to each station for which you code, in the TERMINAL or COMP macro, the CUIDLEN operand (or higher-level configuration macro instruction). The characters will be sent each time the network control program calls the station or answers a call from the station.

If this operand is omitted, the network control program is capable of verifying station ID sequences it receives, but it will not send the controller ID sequence in return.

[NEWNAME= {NCP001 / symbol}]

Specifies the name to be given to the generated network control program load module. Code NEWNAME=symbol, where *symbol* is any valid symbol that does not exceed *seven* characters. (The generation procedure automatically assigns the name you specify, followed by the letter R, to the resource resolution table load module that corresponds to the network control program load module.) Alternatively, code NEWNAME=NCP001, or omit the operand. The network control program load module name will then be NCP001; the resource resolution table load module name will be NCP001R.

[OLT={YES / NO}]

Specifies whether or not the optional online terminal test facility is to be included in the network control program. Code OLT=NO to omit the facility; code OLT=YES (or omit the operand) to include it.

[QUALIFY= {symbol / NONE / SYS1}]

Specifies the first-level qualifier for data sets specified by the NCPLIB, OBJLIB, USERLIB, 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 previously mentioned operands.

symbol

Specifies the qualifier as from one to eight alphanumeric characters; the first character must be aphabetic (including $, @ and #).

NONE

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

SYS1

Specifies that SYS1 is to be used as the qualifier.

```
[RESET= { NONE                                                    }  ]
        { ([IMM  ][,COND][,DEVQ][,OLT][,BLOCK])                   }
        { ALL                                                     }
```

Specifies which of the optional reset functions are to be included in the network control program. Code RESET=NONE (or omit the operand) if none of the options are to be included. Code RESET=ALL if all are to be included.

The remaining parameters indicate specific options to be included:

IMM	Reset Immediate function
COND	Reset Conditional function
DEVQ	Reset Device Queue function
OLT	Reset Online Terminal Test function
BLOCK	Reset Block function

```
[SECCHAN= {INACTIVE}  ]
          {BACKUP  }
```

Specifies the use of the secondary channel.

INACTIVE

Specifies that the secondary channel is installed but is not used by the network control program.

BACKUP

Specifies that the secondary channel is to be used as backup for the primary channel (i.e., it is to be used only if the primary is unavailable). This mode of operation is designed to support loosely coupled processors.

Note: *This operand is valid only if CHANTYP=(TYPE2,TYPE2). If CHANTYP specifies a mixture of channel types, it is assumed that the secondary channel is inactive.*

```
[SLODOWN= {12} ]
         {25}
         {50}
```

Specifies the minimum percent of network control program buffers that are available (that is, not in use) before the network control program enters slowdown mode. When the percent of buffers still available drops below this value, the program reduces the amount of data it accepts from the teleprocessing network and from the host processor, but continues to send to the network and the processor. This procedure reduces the number of buffers in use.

If you omit this operand, the network control program enters slowdown mode when less than 12 percent of the buffers are available.

```
[TRACE= {NO                          } ]
        {(YES[,  {size}])            }
        {        {10  }              }
```

Specifies whether or not the address trace option is to be included in the network control program. Code TRACE=YES to include the option; code TRACE=NO (or omit the operand) to omit the option.

If you code TRACE=YES, you may also specify the number of 16-byte entries the trace table is to contain. Example: TRACE=(YES,20). The minimum number of entries is ten; if you omit the number or specify less than ten, the table will contain ten entries.

```
[TTYCTL={STANDARD} ]
        {SPECIAL  }
```

Specifies how the network control program is to handle the line control sequences when communicating with an 83B3 or a 115A device.

STANDARD
: In the system, the STANDARD support is provided. Specifies that the EOA and EOT sequences are to be removed from the beginning and end of received text and are to be added to the beginning and end of text to be transmitted. In addition, should FIGS H appear within the text, the network control program will transmit FIGS H FIGS so that the receiving device will not consider the sequence as an ending sequence.

SPECIAL
: Specifies that the STANDARD support is not to be included in network control program, that is, EOA and EOT sequences will not be removed from or added to text. Also, the additional FIGS character will not be inserted after a FIGS H sequence.

If this operand is omitted and either 83B3 or 115A devices are included in the system, the STANDARD support is provided.

[TWXCHK={YES / NO}]

Specifies whether or not the network control program will check the parity of data received from TWX 33/35 devices. This operand has no meaning if TWX devices are not included in the system.

Note: *This operand does not affect the handling of parity of data transmitted by the network control program.*

[TWXID=chars]

Specifies the EBCDIC ID Answerback sequence, in hexadecimal representation, for all TWX 33/35 devices in the system. A maximum of 24 hexadecimal characters may be specified. This operand has no meaning if TWX devices are not included in the system.

[USERLIB=dsname]

Specifies the name of a data set that contains the user-written translation tables. (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.) This data set must be cataloged.

If you omit this operand, the generation procedure will assume that any user-written translation tables reside in the data set containing the IBM-supplied modules.

[UT1=dsname]

Specifies the name of a sequential 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.) 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 UNIT=SYSSQ.

[UT2=dsname]

Specifies the name of a sequential 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.) 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 UNIT=SYSSQ.

`[UT3=dsname]`

Specifies the name of a sequential 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.) This data set must be preallocated and cataloged.

If you omit this operand, temporary data sets will be created during each assembly step using UNIT=SYSSQ and each linkage edit step using UNIT=SYSDA.

`[WTTYCTL= {STANDARD} ]`
`          {SPECIAL }`

Specifies how the network control program is to handle the text line control sequences for all WTTY devices in the system.

STANDARD
: Specifies that the EOB, EOT, and WRU sequences are to be removed from and added to the beginning and end of received and transmitted text. In addition, should FIGS H appear within the text, the network control program will transmit FIGS H FIGS so that the receiving device will not consider the sequence as an ending sequence.

SPECIAL
: Specifies that the STANDARD support is not to be included in the network control program, that is, EOB, EOT, and WRU sequences will not be removed from or added to text. Also, the additional FIGS will not be inserted after a FIGS H sequence.

If this operand is omitted and WTTY devices are included in the system, the STANDARD support is provided.

`[WTTYID=chars]`

Specifies the EBCDIC ID Answerback sequence, in hexadecimal representation, to be sent in response to WRU from any WTTY Telex device. A maximum of 40 hexadecimal characters may be specified. This operand has no meaning if WTTY Telex devices are not included in the system.

SYSCNTRL Macro Instruction

SYSCNTRL specifies which of the optional dynamic control facilities are to be included in the network control program. These facilities allow the network control program to execute requests from the host processor to change certain network control program parameters or to determine the status of resources such as lines and stations.

If you omit this macro instruction, the network control program will include none of the dynamic control facilities.

Name	Operation	Operands
[symbol]	SYSCNTRL	[ACTION= {INCLUDE / EXCLUDE}] [,OPTIONS= {ALL / (entry,...)}]

[symbol]

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

[ACTION= {INCLUDE / EXCLUDE}]

Specifies whether the facilities specified by the OPTIONS operand are to be included in or excluded from the network control program. You may list in the OPTIONS operand the facilities to be included, and also codeACTION=INCLUDE. Or you may list in the OPTIONS operand the facilities to be excluded, in which case you would code ACTION=EXCLUDE; other facilities will be included.

[OPTIONS= {ALL / (entry,...)}]

Specifies which of the optional facilities are to be included in or excluded from the network control program.

Code OPTIONS=ALL if you wish all facilities to be included or excluded.

Code OPTIONS=(entry,...) if you wish only certain facilities to be included or excluded; *entry* may be any value listed in Figure 3.

Entry	*Facility*
ACTDV	Activate device
ACTG	Activate group
ACTI	Activate Invites
BHSASSC	Modify block handler set association
CHANL	Switch channel adapters
DATIME	Change date and time
DEACTDV	Deactivate device

Figure 3. Dynamic Control Facilities (Part 1 of 2)

DEACTGF	Deactivate group flush
DEACTLF	Deactivate line flush
DEACTLH	Deactivate line halt
DIAL	Copy or replace station telephone number
DVSTAT	Request device status
LNSTAT	Request line status
LTRACE	Line trace
NAKLIM	Change negative polling response limit
RETCNT	Change retry counts
SESINIT	Copy or replace session initiation information
SESSION	Change session limit
SSPAUSE	Change service seeking pause
STORDSP	Display contents of controller storage
XMTLMT	Change device transmission limit

Figure 3. Dynamic Control Facilities (Part 2 of 2)

Configuration Definition Macro Instructions

HOST Macro Instruction

The HOST macro instruction specifies:

- The number of network control program buffers to be allocated for receiving a data transfer from the teleprocessing access method
- The size of buffer unit in the teleprocessing access method
- The maximum number of buffer units the host processor will allocate for receiving a data transfer
- The number of bytes in the buffer header prefixes used by the teleprocessing access method
- The amount of time the network control program waits for a response by the host processor to an Attention signal.
- Whether the network control program is to return the text portion of a request to fulfill the request.

Name	Operation	Operands
[symbol]	HOST	[,DELAY={count / 0}] INBFRS=count, MAXBFRU=count, UNITSZ=length [,BFRPAD=([{first buffer pad / 0}][,{subsequent buffer pad / 0}])] [,TEXTRET {NONE / BLOCK / ALL}] [,TIMEOUT={count / NONE}]

[symbol]

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

[DELAY={count / 0}]

Specifies the interval, to the nearest tenth of a second, the network control program will delay between the time the network control program has data available for the host processor and the time the network control program presents an Attention to the host processor.

count

Specifies the delay, to the nearest tenth of a second. The minimum delay is 0 seconds (i.e., no delay) and the maximum is 6553.5 seconds.

<u>0</u>

Specifies that an Attention is to be presented to the host processor as soon as data is available.

Note 1: *If the amount of data is sufficient to fill the buffers allocated by the host processor, the Attention will be presented before the delay count has been reached.*

Note 2: *This operand is valid only if CHANTYP=TYPE2 is specified in the BUILD macro.*

`INBFRS=count`

Specifies the number of controller buffers initially allocated for each data transfer to be received from the host processor.

This operand is required, and cannot specify 0.

`MAXBFRU=count`

Specifies the maximum number of buffer units the host processor will allocate for receiving a block of data from the controller.

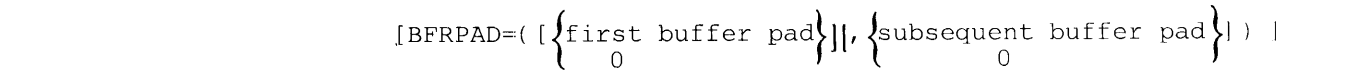
This operand is required.

`UNITSZ=length`

Specifies the size of the buffer units used by the teleprocessing access method. (Specify the actual number of bytes available for holding message data including the pad characters, if any; ignore bytes used for control purposes, such as link fields).

The teleprocessing access method must use one buffer unit size for all transfers between the host processor and the controller. A buffer unit is the smallest unit of contiguous storage handled as buffer space; a buffer may consist of one or more units.

This operand is required.



`first buffer pad`

Specifies the number of pad characters the network control program is to transmit to the *first* access method buffer to be filled with response data. The network control program sends the sequence immediately preceding the response control information. The maximum you may specify is 256.

subsequent buffer pad

Specifies the number of pad characters the network control program is to transmit to all *subsequent* access method buffers in which response data is received. The network control program first sends the pad sequence, then the response data. The maximum you may specify is 256. Placing pad characters at the beginning of each access method buffer allows the access method to insert data, typically message header and message text prefixes. The values you specify in the BFRPAD operand should equal the size of the prefixes, if any, that the access method uses.

$$[\text{TEXTRET}= \left\{ \begin{matrix} \underline{\text{NONE}} \\ \underline{\text{BLOCK}} \\ \text{ALL} \end{matrix} \right\}]$$

Specifies whether or not the text portion of a request is to be returned to the host processor when the network control program cannot successfully transmit the text to the station for which it is intended. Return of text occurs only after all error recovery attempts have been exhausted.

NONE

Specifies that the network control program is not to return the text.

BLOCK

Specifies that the network control program is to return only the block unsuccessfully transmitted. The network control program will not return any blocks it transmitted successfully.

ALL

Specifies that all blocks in the request are to be returned if any block was unsuccessfully transmitted.

$$[\text{TIMEOUT}= \left\{ \begin{matrix} \text{count} \\ \underline{\text{NONE}} \end{matrix} \right\}]$$

Specifies the interval in seconds that the network control program allows to elapse between sending an Attention signal to the host processor via the primary channel and receiving a response to that Attention signal. If this interval expires, the network control program performs one of three actions.

1. If the controller is equipped with a second channel (SECCHAN=YES in the BUILD macro), the network control program sends an Attention signal to the second channel, but continues to wait for a response from the primary channel.
2. If the controller is not equipped with a second channel, and automatic network shutdown is specified (ANS=YES in the BUILD macro), the network control program automatically shuts down the teleprocessing network.
3. If the controller is not equipped with a second channel and you do not include automatic network shutdown in the network control program, the network control program continues to await a response to the Attention signal. No timeout occurs, even if you code a value in the TIMEOUT operand.

`count`

Specifies the interval, expressed as an integral number of seconds (for example, TIMEOUT=15), or to the nearest tenth of a second (TIMEOUT=12.5).

The minimum value is .2 second (TIMEOUT=.2); the maximum is 6553.5 (TIMEOUT=6553.5).

`NONE`

Specifies that the network control program is to wait indefinitely for a response from the primary channel.

The value of TIMEOUT is not counted as one of the 16 possible time intervals permitted in the network control program. (The limit of 16 applies only to communication line timeouts and intervals.)

CSB Macro Instruction

The CSB macro specifies:

- The type of communication scanner
- The internal bit clock rates for the scanner
- The location of the scanner

Name	Operation	Operands
[symbol]	CSB	SPEED=(rate,...) [,MOD={n | 0}] [,TYPE={TYPE1 | TYPE2}]

[symbol]

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

SPEED=(rate,...)

Specifies the internal bit clock rates for up to four oscillators installed in the communication scanner. The speeds must be specified in the same order that the oscillators are installed on the scanner. Standard data rates are shown in Figure 4.

This operand is required.

[MOD={n | 0}]

Specifies whether the communication scanner is located in the 3705 model An or in the 3705 models Bn, Cn, and Dn as follows:

n	*Model*	*Line Interface Addresses (hex)*	
		Type 2	Type 1
0	A1—A2	020-05F	000-03F
1	B1—B4	0A0-0FF	-
2	C1—C6	120-17F	-
3	D1—D8	1A0-1FF	-

[TYPE={TYPE1 | TYPE2}]

Specifies whether the communication scanner is Type 1 or Type 2.

TYPE1

Specifies that the communication scanner is Type 1. TYPE1 is valid only if the MOD operand specifies 0.

TYPE2

Specifies that the communication scanner is Type 2.

Rate	*Represents:*
45	45.5 bps
56	56.89
74	74.2
75	75.0
110	110.0
134	134.5
150	150.0
600	600.0
1200	1200.0
2000	2000.0
2400	2400.0

Figure 4. Standard Communication Scanner Oscillator Bit Rates

IDLIST Macro Instruction

The IDLIST macro specifies:

- A list of identification sequences for BSC stations that call the controller over a switched line.
- The action the network control program is to perform if it receives (from a BSC station) an ID sequence that does not match any sequence in the list.

An identification list is optional for each line over which BSC stations will call the controller (i.e., each line for which CALL=IN or CALL=INOUT is coded in the LINE macro). If you provide a list, the network control program will check ID sequences it receives from a station against the sequences in the list. If you do not provide a list, the network control program will not check the sequences.

Depending on how you code the IDSEQ operand, the network control program will recognize a received ID sequence, or it may also determine which particular station has called.

Name	Operation	Operands
[symbol]	IDLIST	IDSEQ={((chars,term name),...) {(chars,...)} [,MAXLEN=count] [,NOMATCH={PASS {STOP}]

[symbol]

Is any valid symbol. It provides a name for the macro and is referred to by the IDSEQ operand of the LINE macro. A name is required except as indicated in the note below.

[IDSEQ={((chars,termname),...)
{(chars,...)}

Specifies the identification sequence for each of the stations that may call the controller, and optionally, the name of the TERMINAL macro for the station from which the sequence is expected.

chars

Specifies one identification sequence that the network control program is to recognize as valid. You may specify a maximum of 15 characters in one sequence.

[term name]

Specifies the name of the TERMINAL macro representing the station with which the sequence coded as *chars* is to be associated.

If you omit *term name*, the network control program will recognize the sequence as valid, but will not recognize it as the sequence for a specific station.

Note: *You may code a maximum of 255 characters in the IDSEQ operand, including the beginning and ending parentheses and all commas. This limit applies regardless of how many entries you code within the operand. If you need to specify more than 255 characters, code one or more additional IDLIST macros (omitting the name field of each) directly following the first IDLIST macro. In the IDSEQ operand of each additional macro, code the excess entries. (Only the first IDLIST macro may include the MAXLEN and NOMATCH operands.)*

[MAXLEN=count]

Specifies the maximum size of the list, in bytes (*not* the number of entries). This value includes the total bytes in all entries, plus the control fields that precede the list and the individual entries. This operand should be specified only if the teleprocessing access method uses the dynamic control facility either to add entries to the list or to increase the size of one or more existing entries in the list. (The size of an entry increases if a given ID sequence is replaced by a longer sequence.)

The maximum number of ID sequences (entries) the list may contain is 256.

[NOMATCH= {PASS / STOP}]

Specifies the action the network control program is to perform if it receives an ID sequence not recognized as valid (i.e., a sequence not defined in this IDLIST macro), or if it does not receive an ID sequence from the station.

If you wish the network control program to send to the host processor any ID sequences not recognized as valid, code NOMATCH=PASS (or omit the operand). (If the network control program does not receive an ID sequence, this is indicated by the response returned to the access method.

If you do not wish the network control program to send unrecognized sequences to the host processor, code NOMATCH=STOP. Upon receiving an ID sequence it does not recognize, the network control program ignores the line connection, and neither sends data to nor receives data from the station.

LINELIST Macro Instruction

The LINELIST macro defines a list of lines making up a logical line group. This allows the teleprocessing access method to simultaneously activate or deactivate any desired combination of lines, regardless of which physical line groups they appear in. To do so, the access method sends a single control request identifying the logical group to be activated or deactivated.

Any line may be included in any number of logical line groups, and each group is defined by a separate LINELIST macro.

Name	Operation	Operands
symbol	LINELIST	LINES=(entry,...)

symbol

Is any valid symbol. It provides a name for the logical line group.

LINES=(entry,...)

Specifies the name(s) appearing in the name field of one or more LINE macros.

You may code a maximum of 255 characters in this operand, including the beginning and ending parentheses and all commas. This limit applies regardless of how many entries you code within the operand.

SERVICE Macro Instruction

The SERVICE macro generates the service order table for a nonswitched multipoint or nonswitched point-to-point line controlled through a multipoint discipline. A service order table is required for each nonswitched point-to-point line to which an IBM 1050 terminal is connected, and for each nonswitched multipoint line. The SERVICE macro must immediately follow the LINE macro for the line for which the table is being generated.

Name	Operation	Operands
[symbol]	SERVICE	ORDER=(entry,...) [,MAXLIST=n]

[symbol]

Is any valid symbol. It provides a name for the service order table.

ORDER=(entry,...)

Specifies the order in which the devices on the communication line are to be serviced.

entry is the name of a station or component to be serviced by the network control program. These names appear on the TERMINAL, CLUSTER, and COMP macros associated with the line for which the service order table is being generated.

The maximum number of entries is 256.

Note: *You may code a maximum of 255 characters in the ORDER operand, including the beginning and ending parentheses and all commas. This limit applies regardless of how many entries you code within the operand. If you need to specify more than 255 characters, code one or more additional SERVICE macros following the first SERVICE macro. In the ORDER operand of each additional macro, code the excess entries.*

[MAXLIST=n]

Specifies the maximum number of entries in the service order table. The maximum value for n is 256. If the number you specify in MAXLIST exceeds the number of entries you code in the ORDER operand, you may add more entries (up to the MAXLIST limit) during network control program execution. The teleprocessing access method sends a control request to add entries.

If this operand is omitted, *n* is assumed to equal the number of entries in the ORDER operand, and no further entries may be added during network control program execution. (Entries may be *changed*, however, with the dynamic control function.)

LCST Macro Instruction

The LCST macro defines one entry in the line control selection table. This table provides information the network control program needs to communicate with a station that calls the controller over a switched line defined as a multiple-terminal-access line.

Note: *All LCST macro instructions must be grouped together, and MTALIST macros must immediately follow the LCST instructions.*

If any line in the network is a multiple-terminal-access line, a line control selection table is required. Only one table is required regardless of the number of such lines. Each entry in the table represents a particular combination of parameters associated with a terminal. These parameters include the transmission code and the type of line control to be used.

The maximum number of LCST macros you may code is 64.

Name	Operation	Operands
symbol	LCST	GROUP=entry [,CODE= {BCD / EBCD / COR}] [,LCTYPE= {1050 / 2740A / 2740D / 2740E / 2740F / 2741}]

symbol

Is any valid symbol. It provides a name for this entry in the line control selection table and is required; the name is referred to by the LCST operand of the MTALIST macro.

GROUP=entry

Specifies the name of the GROUP macro for any line group whose station characteristics are the same as the characters of the station that will call the controller over the multiple-terminal-access line; (*entry* must not be the name of a GROUP macro in which TERM=MTA is coded).

[CODE= {BCD / EBCD / COR}]

Specifies the transmission code with which the network control program is to communicate with the type of terminal represented by this macro.

Valid transmission codes for each type of terminal are as follows. (Asterisks indicate the code assumed if you omit this operand.)

```
                CODE=
IBM 1050:       EBCD* (Extended BCD code)
                BCD   (BCD code)

IBM 2740:       EBCD*
                BCD
                COR   (Correspondence code)

IBM 2741:       COR*
                BCD
                EBCD
```

[LCTYPE={1050 | 2740A | 2740D | 2740E | 2740F | 2741}] |

Specifies the type of terminal line control that this entry represents, as follows:

Entry	*Type*
1050	IBM 1050
2740A	IBM 2740 basic
2740D	IBM 2740 with Transmit Control feature
2740E	IBM 2740 with Transmit Control and Checking features
2740F	IBM 2740 with Checking feature
2741	IBM 2741

MTALIST Macro Instruction

The MTALIST macro provides a list used by the network control program to determine the *type* of terminal, when a terminal calls the controller over a multiple-terminal-access line. MTALIST macros must be coded immediately after the LCST instructions.

A multiple terminal access list is required for each line defined as a multiple-terminal-access line. The same list may be used for more than one MTA line, if the terminal characteristics are the same for all the lines.

Name	Operation	Operands
[symbol]	MTALIST	LCST=(lcst name,...)

[symbol]

Is any valid symbol. It provides a name for the Multiple Terminal Access List; the name is referred to by the MTALIST operand of the LINE macro. A name is required except as indicated in note 1 below.

LCST=(lcst name,...)

Specifies a list of line control selection table entries. *lcst name* is the name of a LCST macro.

Include in this operand one entry for each kind of terminal that will call the controller over the multiple-terminal-access line represented by this MTALIST macro. (*Kind* of terminal does not refer solely to the type [1050, 2740, 2741], but encompasses the transmission code, timeout values, and terminal features as well.)

The maximum number of entries is 64.

Notes:

1. *You may specify a maximum of 255 characters in the LCST operand, including the beginning and ending parentheses and all commas. This limit applies regardless of how many entries you code within the operand. If you need to specify more than 255 characters to complete the list, code one or more additional MTALIST macros (omitting the name field of each) directly following the first MTALIST macro for this list. In the LCST operand of each, code the excess entries.*
2. *All MTALIST macros, if any, must follow the last LCST macro in the network control program generation input statements.*

MTAPOLL Macro Instruction

The MTAPOLL macro instruction specifies all of the polling characters used by all IBM 1050 terminals that may call the controller over any multiple-terminal-access line. Both common and specific polling characters may be specified.

Only polling characters for 1050 terminals that will call the controller over a multiple-terminal-access line need be specified.

Only one MTAPOLL macro may be specified in the network control program generation input statements.

Name	Operation	Operand
[symbol]	MTAPOLL	POLL=(chars,...)

[symbol]

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

POLL=(chars,...)

Specifies the list of polling characters.

DIALSET Macro Instruction

The DIALSET macro instruction specifies the switched point-to-point lines that are to make up a dial set. (A dial set is the group of lines from which the network control program selects a line over which to call a station.) A dial set may contain any number of lines, but all must have similar characteristics. This allows the network control program to use any of the lines to call a station of a specific type.

Name	Operation	Operand
[symbol]	DIALSET	LINES=(line name,...) [,DIALALT=dialset name] [,QLIMIT={count / 1}] [,QLOAD={count / 0}] [,RESERVE={count / 0}]

[symbol]

Is any valid symbol. It provides a name for the dial set and is required except as indicated in the note following the LINES operand. The name is referred to by the DIALSET operands of the LINE and TERMINAL macros and the DIALALT operands of the LINE and DIALSET macros.

LINES=(line name,...)

Specifies the switched lines of which the dial set is to consist; *line name* is the name of the LINE macro for a line to be included. Only a line whose LINE macro specifies CALL=OUT or CALL=INOUT may be included in a dial set. No line may appear in more than one dial set.

This operand is required.

Note: *You may specify a maximum of 255 characters in the LINES operand, including the beginning and ending parentheses and all commas. This limit applies regardless of how many line names you code within the operand. If you need to specify more than 255 characters to complete the list, code one or more additional DIALSET macros (omitting the name field of each) after the first DIALSET macro for this dial set. In the LINES operand of each, code the excess line names.*

[DIALALT=dialset name]

Specifies the name of a dial set that is to serve as an alternate dial set. *dialset name* is the name of the DIALSET macro for the alternate dial set. The alternate dial set must consist of the same type of lines as the primary dial set. Then the network control program can communicate with the stations with which the primary dial set is associated.

Note: *The DIALSET macro specified by the DIALALT macro must immediately follow the DIALSET macros defining this dial set. Omit the DIALALT operand from the last of a chain of DIALSET macros; do not use the operand to specify the name of the first DIALSET macro in the chain.*

Correct:

```
DS1     DIALSET     LINES=(...),DIALALT=DS2
DS2     DIALSET     LINES=(...),DIALALT=DS3
DS3     DIALSET     LINES=(...)
```

(The third macro would be incorrect if it specified DIALALT=DS1.)

[QLIMIT={count / 1}]

Specifies the maximum number of requests the network control program will allow to accumulate on the queue for the dial set. When this limit is reached, the network control program rejects (returns to the host processor) any further call-out requests it receives. Rejection will continue until the number of requests becomes less than the queue limit value specified. The minimum queue limit is one; the maximum is 255.

[QLOAD={count / 0}]

Specifies the number of call-out requests the network control program will allow to accumulate on the queue for this dial set before using a line from the alternate dial set to call the station. If you code QLOAD=0 (or omit the operand), and an alternate dial set is specified via the DIALALT operand, the network control program uses a line from the alternate dial set if no line in the primary dial set is available.

The maximum number of requests you may specify is 255. The value specified in QLOAD must be less than the value in QLIMIT; otherwise the queue of unfulfilled call-out requests cannot reach the size that will cause the network control program to use the alternate dial set.

[RESERVE={count / 0}]

Specifies the number of lines in the dial set to be reserved for incoming calls from stations. If you code RESERVE=0 (or omit the operand), the network control program will reserve no lines; when all lines are busy with outgoing calls, no stations will be able to call the communications controller.

The maximum value for RESERVE is 255.

TRANSTBL Macro Instruction

The TRANSTBL macro provides for modification of the standard code translation tables.

CAUTION
Diagnostic aids such as online tests are designed for use with the standard, IBM-provided translation tables. Any changes made to the standard tables may prevent successful operation of these aids by introducing characters that the test routines do not recognize as valid. This fact should be considered when modifying translation tables. Any changes should be documented and made known to the IBM customer engineering personnel.

Name	Operation	Operand
symbol	TRANSTBL	CODE=type [,EOB={WRU / FIGSchars / chars}] [,EOT= chars] [,IAM={YES / NO}] [,FGSLTRS={IN / OUT}] [,SPSHIFT={YES / NO}] [,TYPE={XMIT / RCV}] [,Xeeoo=([hexchar1][,hexchar2])]

symbol

Any valid symbol. Provides a name for the translation table. The name is required. (Referenced by the TRANS operand on the LINE macro.)

CODE=type

Specifies the name of a standard translation table to be modified.

[EOB={WRU / FIGSchars / chars}]

Specifies the logical EOB sequence for WTTY devices.

WRU
Specifies that the WRU (i.e., FIGS D) sequence is to be considered as an EOB sequence.

FIGSchars
Specifies that the FIGS character followed by 'chars' is to be considered as an EOB sequence. 'chars' is specified as any two hexadecimal characters.

chars
Specifies an EBCDIC character, in hexadecimal representation. The occurrence of four consecutive 'chars' is considered as an EOB sequence.

[EOT=chars]

Specifies an EBCDIC character, in hexadecimal representation. The occurrence of FIGS chars LTRS is considered as a logical EOT sequence for WTTY devices.

[IAM={YES / NO}]

Specifies whether or not the network control program is to recognize the WRU, (i.e., FIGS D) sequence on input. If IAM=YES and the WRU sequence is received, the network control program will send the WTTYID specified on BUILD.

[FGSLTRS={IN / OUT}]

Specifies whether the FIGS and LTRS characters are to be left in or taken out of text received from 83B3 or 115A devices.

[SPSHIFT={YES / NO}]

Specifies whether or not the space character is to be considered as a downshift for 83B3 or 115A devices.

[TYPE={XMIT / RCV}]

Specifies whether the table specified by CODE is to be used for translating data being transmitted to (XMIT) or received from (RCV) the devices.

[,Xeeoo=([hexchar1][,hexchar2])]

Specifies changes to be made to the standard translation table specified by the CODE operand. Xeeoo identifies two characters; one or both may be modified. 'eeoo' is an even odd pair which may assume the values 0001-FEFF. 'ee' and 'oo' are the hexadecimal representation of the characters to be translated. The characters will be translated to the values specified by the corresponding hexchar1 and hexchar2.

hexchar1
Specifies the hexadecimal representation of the character to which 'ee'is to be translated. If hexchar1 is omitted, the standard translation of the character 'ee' will be unchanged.

hexchar2
Specifies the hexadecimal representation of the character to which 'oo'is to be translated. If hexchar2 is omitted, the standard translation of the character 'oo' will be unchanged.

Note 1: *Xeeoo need only be specified for those characters which require a different translation than that provided by the standard translation table. The Xeeoo operand may be specified as many times as is necessary to obtain the desired translation.*

Note 2: *The line control characters may not be changed in the translation table.*

Teleprocessing Network Configuration Macro Instructions

A teleprocessing network configuration may be viewed as a logical arrangement of elements, each kind of element occupying a different level. These kinds of elements are, in decreasing order, *lines, stations*, and station *components.*

Consider a network containing three lines, six stations, and seventeen components, arranged as shown in Figure 5. Assume that each component is to be separately pollable or addressable.

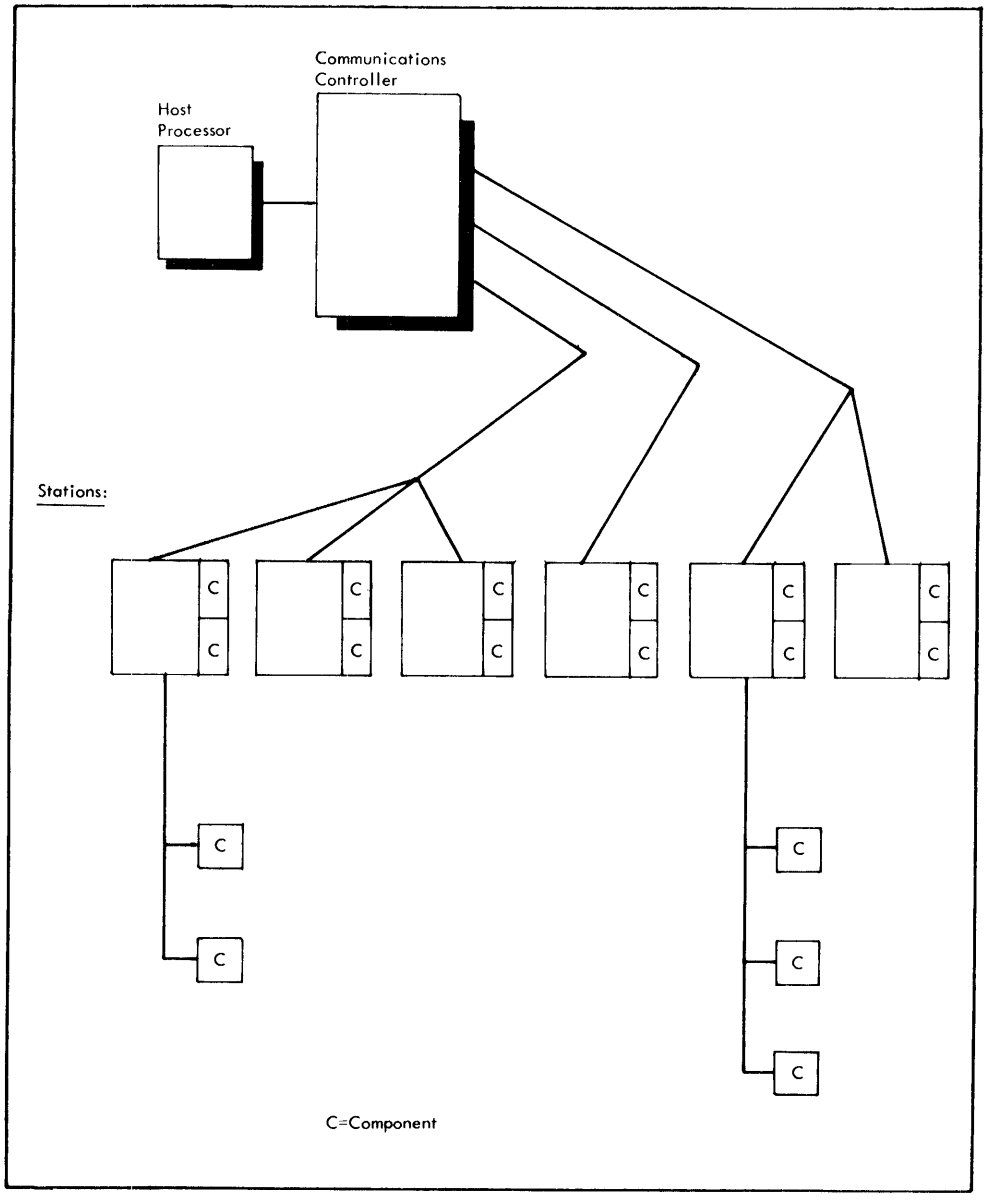


Figure 5. Example of Network Configuration

The corresponding macro instructions for this configuration appear as follows:

```
LINE
    TERMINAL
        COMP
    TERMINAL
    TERMINAL
LINE
    TERMINAL
LINE
    TERMINAL
        COMP
        COMP
    TERMINAL
```

(The staggered arrangement of the macros is intended only to illustrate the different levels; they need not be indented as shown).

In the example, the terminals having only two components (one input and one output) are assumed to be polled and addressed using the general polling and station addressing sequences specified in the TERMINAL macro. No COMP macros are shown for these terminals. Those terminals for which extra input and output components are shown are assumed to require individual polling and addressing. COMP macros are therefore shown for these terminals. (In many applications, individual polling and addressing of components is unnecessary, and the TERMINAL macro may represent all of the components. The POLL and ADDR operands specify the general polling and station addressing sequences respectively. Avoiding use of COMP macros reduces the amount of storage required for the network control program control tables.

One or more GROUP macros must be added. The GROUP macro does not represent an element of the network as do the LINE, TERMINAL, and COMP macros. It 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. (Contrast with *logical* line group.)

Each LINE macro in the macro sequence must be associated with a preceding GROUP macro. Assume in the example the first two lines have similar attributes allowing them to be in the same line group, but the third line must be in its own group. The macro sequence becomes:

```
GROUP
    LINE
        TERMINAL
            COMP
        TERMINAL
        TERMINAL
    LINE
        TERMINAL
GROUP
    LINE
        TERMINAL
            COMP
            COMP
        TERMINAL
```

If all the lines were dissimilar, each of the three LINE macros would have to be preceded by a GROUP macro. If all were alike, all lines could be in the same line group, with only one GROUP macro immediately preceding the first LINE macro. (An exception is the SERVICE macro, which must appear directly after the LINE macro if the line is a multipoint line, or a nonswitched point-to-point line to which an IBM 1050 is connected.)

Clustered Stations

The preceding arrangement differs for *clustered* stations (that is, the IBM 2972 General Banking Terminal System and the IBM 3270 Information Display System). For these stations the sequence of configuration macros is LINE, CLUSTER, and TERMINAL, instead of LINE, TERMINAL, and COMP.

For example, consider a line connected to two terminal control units (e.g., 2972). Three terminals (e.g., 2980) are connected to each of these control units. The line is represented by the sequence:

```
LINE
   CLUSTER
      TERMINAL
      TERMINAL
      TERMINAL
   CLUSTER
      TERMINAL
      TERMINAL
      TERMINAL
```

The lowest level is represented by the TERMINAL macros; the stations (terminal control units) are represented by CLUSTER macros.

Advantage of Macro Sequencing

The principal advantage of macro sequencing is that it saves coding effort. For example, each of five characteristics common to all of fifteen terminals on a line can be specified once, in the LINE macro, rather than in each of the fifteen individual TERMINAL macros. Five operands are coded instead of 75 (15 x 5).

The characteristic need not be identical for *all* of the elements at a level to specify it at a higher level. You may code the exceptions at the lower level; any characteristic you code at the lower level automatically overrides the characteristic specified at the higher level.

If, for example, only twelve of the fifteen terminals mentioned previously have the same characteristic in common, you could still specify that characteristic in the LINE macro; then you would specify the differing characteristics in the TERMINAL macros for the three exceptional terminals.

Figure 6 provides a summary of the operands of each configuration macro instruction and indicates the operands each macro can contain. The rightmost bullet (•) indicates the lowest-level macro in which that operand can be coded. The description of the operand is given under the lowest-level macro, except all of the operands of the COMP macro are described under the TERMINAL macro.

Operand:	Macro Instructions				
	GROUP	LINE	CLUSTER	TERMINAL	COMP
ADDR				•	•
ADDRESS		•			
ATTN	•	•		•	
AUTO	•	•			
BFRDLAY	•	•	•	•	
BHEXEC	•	•	•	•	•
BHSET	•	•	•	•	•
CALL	•	•			
CLOCKNG	•	•			
CODE	•	•			
CRITSIT	•	•	•	•	
CTERM				•	
CUIDLEN	•	•		•	
CUTOFF	•	•			
CUTYPE	•	•	•		
DATRATE	•	•			
DIAL	•				
DIALALT	•	•			
DIALNO				•	
DIALSET	•	•		•	
DISCNCT	•	•			
DUPLEX	•	•			
ENDTRNS	•	•		•	•
FEATURE[1]	•	•	•	•	
GPOLL			•		
IDSEQ	•	•			
INTPRI	•	•			
ISTATUS	•	•	•	•	•
LNCTL	•				
MTALIST	•	•			
PADCNT	•	•			
PAUSE	•	•			
POLIMIT	•	•			
POLL				•	•
POLLED	•	•			

[1]Only BATCH or NOBATCH may be specified in the FEATURE operand of the CLUSTER macro, for the IBM 2972 General Banking Terminal System.

Figure 6. Summary of Operands for Configuration Macro Instructions (Part 1 of 2)

Operand:	Macro Instructions				
	GROUP	LINE	CLUSTER	TERMINAL	COMP
RECORD	•	•	•	•	•
REDIAL	•	•			
REPLYTO	•				
RETRIES	•	•			
RING	•	•			
SERVLIM	•	•			
SESSION	•	•			
SPEED	•	•			
SUBCHAR	•				
SYNDLAY	•				
TADDR	•	•			
TERM	•	•	•	•	
TEXTTO	•				
TRANS	•	•			
TRANSFR	•	•			
TTDCNT	•				
WACKCNT	•				
WAKDLAY	•				
XMITLIM	•	•	•	•	•
YIELD	•	•			

Figure 6. Summary of Operands for Configuration Macro Instructions (Part 2 of 2)

GROUP Macro Instruction

A communication line group consists of lines that have certain characteristics in common.

All lines in the group are nonswitched (point-to-point or multipoint, or a combination), or all are switched.

All stations connected to lines in the group are start-stop stations, or all are binary synchronous. If binary synchronous, they may be of different types, in any combination—for example, IBM 2770, IBM 2780, and IBM 1130. All BSC stations use a uniform line control scheme.

If the stations are start-stop, they must be of the same type—for example, they may be IBM 1050 or IBM 2741, but not both. If they are IBM 2740 terminals, they must have certain features in common. For example, a line group cannot include both 2740s with the checking feature and 2740s without this feature.

An exception is a line group consisting of multiple-terminal-access lines. These can accommodate IBM 1050, 2740, and 2741 terminals in combination.

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, TERMINAL, and COMP (or LINE, CLUSTER, and TERMINAL) macros for lines and devices within the group, and specifies:

- Whether the lines are switched or nonswitched.
- 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	[,DIAL= {NO / YES}] [,LNCTL= {SS / BSC}] [,REPLYTO= {count / 3.0}] [,SUBCHAR= {char / 3F}] [,SYNDLAY= {count / 1.0}] [,TEXTTO= {count / 23.5}] [,TTDCNT= {count / 15}] [,WACKCNT= {count / 15}] [,WAKDLAY= {count / 2.2}]

symbol

Is any valid symbol. It provides a name for the group and is required.

[DIAL= {NO / YES}]

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

[LNCTL= {SS / BSC}]

Specifies whether the line group contains start-stop lines (LNCTL=SS) or BSC lines (LNCTL=BSC).

[REPLYTO= {count / 3.0}]

Specifies the reply timeout value for the lines in the line group. If the network control program does not receive a response to polling or to message text it has sent to a station, it makes no further attempt to communicate with the station. Instead, it indicates that a timeout error has occurred.

You may specify this value as an integral number of seconds or to the nearest tenth of a second. Reply timeouts for a start-stop line are resolved to the

nearest half second; reply timeouts for BSC lines, to the nearest tenth of a second.

Example: If you specify a value of 4.6 for a BSC line group, the reply timeout value will be 4.6 seconds. If you specify 4.6 for a start-stop line group, the reply timeout value will be 4.5 seconds (nearest half second to 4.6).

The maximum value is 1632 seconds.

For most teleprocessing networks, the standard value of 3 seconds is appropriate. Use of this value is recommended unless the system designer specifies a different one.

```
[SUBCHAR= {char}]
         {3F  }
```

Specifies the character to be substituted in the translated text for a character on which a data check occurred. The substitution character must be specified as the hexadecimal representation of the EBCDIC character without the framing characters X' '. The default value, 3F, is the EBCDIC SUB (substitute) character.

```
[SYNDLAY= {count}]                    (BSC lines only)
          {1    }
```

Specifies the interval between transmissions by the network control program of the BSC synchronizing characters (SYN) on a line when that line is in text-transmit mode.

You may specify this interval as an integral number of seconds or to the nearest tenth of a second.

Example: For an interval of two seconds you code SYNDLAY=2 (or 2.0); for 1.5 seconds, you code SYNDLAY=1.5.

The maximum interval you may specify is 1632 seconds.

For most teleprocessing networks the standard value of 1 second is appropriate. Use of this value is recommended unless the system designer specifies a different one.

```
[TEXTTO= {count}]
         {23.5 }
```

Specifies the text timeout value for the lines in the line group. If the interval between any two successive message characters received from a station exceeds this value, the network control program ends the Read or Invite operation with a text timeout error indication.

You may specify this value as an integral number of seconds or to tenths of a second. Text timeouts for a start-stop line are resolved to the nearest half second; text timeouts for BSC lines, to the nearest tenth of a second.

Example: If you specify a value of 20.2 for a BSC line group, the text timeout value will be 20.2 seconds. If you specify a value of 20.2 for a start-stop line group, the text timeout value will be 20 seconds (nearest half second to 20.2).

The maximum value is 1632 seconds.

For most teleprocessing networks the standard value of 23.5 seconds is appropriate. Use of this value is recommended unless the system designer specifies a different one.

```
[TTDCNT= {count}]          (BSC lines only)
         {15   }
```

Specifies the maximum number of times the BSC temporary text delay (TTD) sequence is to be transmitted to a station before the transmission is aborted. The TTD sequence notifies the station that the controller is temporarily unable to send the next block of data. The maximum count is 255. Any value less than 255 specifies the exact number of times the sequence is to be sent; 255 specifies that the sequence is to be sent without limit.

```
[WACKCNT= {count}]         (BSC lines only)
          {15   }
```

Specifies the maximum number of times the BSC wait-before-transmit (WACK) sequence is to be transmitted to a station before a transmission is to be aborted. The WACK sequence signals to the station that the controller is temporarily not ready to receive. Any value less than 255 specifies the exact number of times the sequence is to be sent; 255 specifies that the sequence is to be sent without limit.

The maximum count is 255.

For most teleprocessing networks, the standard value of 15 WACK transmissions is appropriate. Use of this value is recommended unless the system designer specifies a different one.

```
[WAKDLAY= {count}]         (BSC lines only)
          {2.2  }
```

Specifies the maximum time delay that is to elapse before the network control program responds to message text received from a station on any line in the line group. If the network control program has been unable to respond normally (e.g., with a positive acknowledgment) by the time this interval has elapsed, it will send a WACK sequence instead.

You may specify this delay as an integral number of seconds or to the nearest tenth of a second.

Example: You code a delay of 12 seconds as WAKDLAY=12 (or WAKDLAY=12.0); for 12.5 seconds, you code WAKDLAY=12.5.

The maximum number of seconds you may specify is 1632 (1632.0) seconds.

For most teleprocessing networks the standard interval of 2.2 seconds is appropriate. Use of this value is recommended unless the system designer specifies a different one.

Note: *A maximum of 16 different time intervals may be specified in the network control program. This includes (1) seven standard intervals that are common to all network control programs, (2) intervals that you explicitly specify in the REPLYTO, SYNDLAY, TEXTTO, and WAKDLAY operands of the GROUP macro, and (3) the default values for those operands. If the total number of* different *time intervals exceeds 16, generation of the network control program is terminated. An example of different timeout values is REPLYTO=10, WAKDLAY=4. The* same *time value specified in any number of the four operands mentioned constitutes only a* single *time interval.*

The standard time intervals for all network control programs are:

0.0 (no time interval)
1.0 seconds
2.2 seconds
3.0 seconds
3.5 seconds
23.5 seconds
60.0 seconds
IMM (immediate action required)

In addition to the preceding operands, most operands of the LINE, CLUSTER, TERMINAL, and COMP macros can be specified in the GROUP macro instead of the individual macros mentioned. Figure 6 shows which of the lower-level operands you may specify in the GROUP macro.

LINE Macro Instruction

The LINE macro represents one communications line attached to the communications controller and specifies:

- Whether the line is a half-duplex or full-duplex line
- Whether or not the Auto Call feature is available for the line
- Whether or not the controller is to be considered the primary station on a nonswitched point-to-point BSC contention line
- The line address
- The speed of the line
- The interrupt priority of the line
- Whether the data set or the controller is to provide clocking
- Whether the controller is a tributary station on the line represented by this macro, and if so, the addressing character to which the controller will respond
- Whether (for a switched line) the modem is equipped with the "ring indication interface" lead
- Whether or not the stations on the line are polled
- The transmission code
- Certain procedural options the network control program is to use for the line.

One LINE macro must be coded for each communication line connected to the 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.

Name	Operation	Operands
symbol	LINE	ADDRESS=line address SPEED=rate [,AUTO={C / NONE}] [,CALL={IN / OUT / INOUT}] [,CLOCKNG={INT / EXT}] [,CODE={EBCDIC / USASCII / BCD / EBCD / COR}] [CRCNT={count / 10}] [,CUTOFF={count / NO}] [,DATRATE={HIGH / LOW}] [,DIALALT={entry / NONE}]

```
[,DIALSET= {entry}  ]
           {NONE }

[DISCNCT= {count}  ]
          {2    }

[,DUPLEX={HALF}  ]
         {FULL}

[,IDSEQ= {symbol}  ]
         {NONE  }

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

[,MTALIST=entry]

[PADCNT= {count}  ]
         {0    }

[,PAUSE= {t}  ]
         {0}

[,POLIMIT=( [ {n}  ][,  {WAIT                }  ])]
              {1}       {QUEUE[,NOTIFY]}

[,POLLED={NO }  ]
         {YES}

[,REDIAL={count}  ]
         {3    }

[,RETRIES= {NONE          }  ]
           {(m[,t[,n]])   }

[,RING= {YES}  ]
        {NO }

[,SERVLIM=  count  ]

[,SESSION= {count}  ]
           {1    }

[,TADDR= {char  }  ]
         {NONE  }

[TRANS=([inname][,outname])]

[,TRANSFR= {count}  ]
           {7    }

[,YIELD= {YES}  ]
         {NO }
```

`symbol`

Is any valid symbol. It provides a name for the line and is required.

`ADDRESS=line address`

Specifies the line interface address of this line. Code this as the hexadecimal line address without the framing characters X" (e.g., ADDRESS=02F). As shown under the CSB macro instruction, valid line addresses are in the ranges 020-05F, 0A0-0FF, 120-17F, 1A0-1FF, for a type 2 scanner, or 000-03F, for a type 1 scanner. Determine the address of each line from the system designer.

This operand is required.

`SPEED=rate`

Specifies the initial data rate for this line.

If CLOCKNG=INT, this rate must be one of the four rates specified for the communication scanner to which this line is attached. Specify, as the rate, 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 CLOCKNG=EXT, at least one lower rate must have been specified on the associated CSB macro.

If the rate exceeds 2400 bits per second, CLOCKNG=EXT must be specified.

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

This operand is required.

`[AUTO= {C / NONE}]`

Specifies whether the Auto Call facility is present for this line. This may be determined from the system designer. (Any line that the network control program is to use to call a station (i.e., you code CALL=OUT or CALL=INOUT in this macro) must be equipped with the Auto Call facility.

If the line is so equipped, code AUTO=C; if not, code AUTO=NONE or omit the operand.

`[CALL= {IN / OUT / INOUT} ]`

Specifies whether stations, or the network control program, or both, can initiate calls via the line represented by this LINE macro.

If the line is to be used only for incoming calls, (stations call the controller), code CALL=IN, or omit the operand.

If the line is to be used only for outgoing calls (controller calls stations), code CALL=OUT.

If the line is to be used for both incoming and outgoing calls, code CALL=INOUT.

To specify CALL=OUT or CALL=INOUT in this macro, the line must be equipped with an Auto Call unit, and you must code AUTO=C in this macro.

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

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

If the controller provides clocking, code CLOCKNG=INT. If the modem (data set) provides clocking, code CLOCKNG=EXT, or omit the operand.

```
[CODE= {EBCDIC }    ]
       {USASCII}
       {BCD    }
       {EBCD   }
       {COR    }
```

Specifies the transmission code with which the network control program is to communicate with stations over the line represented by this LINE macro. This applies only to types of stations for which a choice of codes is available. Determine from the system designer the codes used.

Figure 7 lists the codes that may be specified for each type of station. The CODE operand should not be specified for any type of station not shown in Figure 7. (If a code is specified, it is ignored.)

Code	CODE=	Type of Station			
		Any BSC station	IBM 1050	IBM 2740	IBM 2741
Extended Binary Coded Decimal Interchange Code	EBCDIC	•[1]			
USA Standard Code for Information Interchange	USASCII	•			
Extended Binary Coded Decimal	EBCD		•[1]	•[1]	•
Binary Coded Decimal	BCD		•	•	•
Correspondence	COR			•	•[1]

[1] Code assumed if CODE operand is omitted.

Figure 7. Valid CODE Operand Values

[CRCNT= {count / 10}]

Specifies, for WTTY, the number of character positions through which the print element will return during the time required to transmit one character.

[CUTOFF= {count / NO}]

Specifies a maximum number of sub-blocks the network control program will accept from a station, or specifies no limit to the number of sub-blocks. If the network control program receives this number of sub-blocks before receiving an end-of-block character from the station, it breaks off the transmission. (A sub-block is the sequence of message text occupying the number of buffers specified by the TRANSFR operand of this macro.)

If you wish to establish a limit, code CUTOFF=count, where *count* is from 1 to 255. If you do not wish to establish a limit, code CUTOFF=NO or omit the operand. The network control program will continue to accept message text from the station until it receives an end-of-block character.

[DATRATE= {HIGH / LOW}]

Specifies which data rate is to be used on a modem (data set) that has a dual data rate.

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.

[DIALALT= {entry / NONE}]

Specifies the name of a dial set from which the network control program will select a switched line (if available) as an alternate path for a nonswitched point-to-point line.

If the network control program is unable to reach the station connected to the nonswitched line, the program uses a line from the dial set. For the alternate path to be used successfully, the lines in the dial set must have the same characteristics as the nonswitched point-to-point line.

Code a dial set name in this operand only if (1) the line represented by this LINE macro is a nonswitched point-to-point line, (2) the station connected to this line is equipped with the Dual Communications Interface feature, and (3) you wish the network control program to use the alternate path retry option when communication with the station over the nonswitched line is not possible.

[DIALSET= {entry / NONE}]

Specifies the name of the dial set to which this line belongs. *entry* is the name of a DIALSET macro. The LINES operand of the DIALSET macro must contain the name of this LINE macro.

The DIALSET operand must specify the name of a dial set if this LINE macro represents a switched point-to-point line over which the network control program will call stations. (CALL=OUT or CALL=INOUT must also be coded in this LINE macro.)

Code DIALSET=NONE (or omit the operand) if the line is used only for incoming calls (CALL=IN) or if it is not a switched line.

[DISCNCT= {count / 2}]

Specifies the length of continuous space required to break a Telex switched connection for a WTTY line. This operand is valid only if WTTY Telex devices are on the line.

[DUPLEX={HALF / FULL}]

Specifies whether or not the communications line is physically a half-duplex or full-duplex line. Determine from the system designer the appropriate value to code.

Note: *This should not be confused with half-duplex or full-duplex data transfer. This operand specifies only the physical characteristic of the line. (All data transfer between the controller and any of the stations supported by the network control program occurs only in half-duplex mode, regardless of whether the line is half-duplex or full-duplex.)*

```
[IDSEQ= {symbol}  ]
        {NONE  }
```

Specifies whether or not the network control program is to verify identification (ID) sequences received from any BSC station that calls the controller over the switched line represented by this LINE macro.

If you wish the network control program to verify ID sequences, you must specify in an IDLIST macro a list of valid ID sequences against which the network control program is to compare incoming ID sequences. Code IDSEQ=symbol, where symbol is the name of an IDLIST macro.

Code IDSEQ=NONE, or omit the operand, if you do not wish the network control program to verify incoming ID sequences.

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

Specifies the interrupt priority for this line. Priority 3 is highest and 0, lowest. Lines with high data rates should be assigned higher priorities than lines with lower data rates.

If this line is attached to a Type 2 communication 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. Determine from the system designer the appropriate value to code.

```
[MTALIST=entry]
```

Specifies that the line represented by this LINE macro is to be used for multiple terminal access, and identifies the list the network control program will use to identify the type of terminal calling the controller over this line. Entry is the name of an MTALIST macro that defines the list to be used to identify the terminal type.

```
[PADCNT={count}  ]
        {  0  }
```

Specifies the number of idle characters to be sent to a WTTY terminal to permit its motor to come to full speed. This is required only for a line on which WTTY terminals do not have the heavy duty motor option which does not switch off the motor after 10 seconds of inactivity. Sufficient characters must be specified to create a 1.5 second delay on the line.

If you omit this operand, the line represented by this LINE macro cannot be used for multiple terminal access.

```
[PAUSE= {t}  ]
       {0}
```

Specifies the number of seconds of delay between successive service seeking operations when the session limit has not been reached for the line. It may be from 0 to 255; if you omit the operand, 0 is assumed, i.e., there is no delay between successive scans.

```
[POLIMIT=( [{n}] [,{WAIT                }] )]
          [{1}]  {QUEUE[,NOTIFY]}
```

Specifies (1) the number of consecutive negative responses to polling the network control program will accept from a station before breaking off communication; (2) the action the network control program is to take when the limit is reached. The maximum value of *n* is 255.

This limit applies only to polling performed after the network control program has received at least one message block from the station. It does not apply to initial polling.

```
[ {WAIT            }]
  {QUEUE[,NOTIFY]}
```

Specifies the action the network control program is to perform if the maximum number of negative responses, *n*, is reached.

WAIT

Specifies that the logical connection between the network control program and the station is to be maintained. The network control program notifies the host processor that the negative response limit has been reached, then awaits another request from the host processor before performing any further action on that line.

QUEUE[,NOTIFY]

Specifies that the network control program is to break the logical connection with the station and queue the current request onto the beginning of the queue for the station, for later servicing.

[NOTIFY]

Specifies that the network control program is to notify the host processor that the negative response to polling limit has been reached for the station.

If you omit NOTIFY, the network control program does not notify the host processor.

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

Specifies whether or not this line is polled. Code POLLED=YES if the line is a multipoint line or if it is a nonswitched or switched point-to-point line used to communicate with an IBM 1050 terminal.

Code POLLED=NO (or omit the operand) if the line (1) is a nonswitched or switched point-to-point line (unless the terminal is an IBM 1050) or (2) is a multiple-terminal-access line (MTALIST=YES is coded).

Note: *This operand has no meaning if MTALIST is specified and is ignored if coded.*

```
[REDIAL={count}]
        {  3  }
```

Specifies how many times the network control program is to perform the dialing operation in attempting to reach a station over a switched line. The maximum number is 255. (Any value less than 255 specifies the exact number of dialing operations following the initial operation. 255 specifies that the network control program is to dial the station without limit.)

```
[RETRIES= {NONE          }]
          {(m[,t [,n]]) }
```

Specifies the number of attempts, via retransmission, to recover from text errors in message data. There can be one or more sequences of consecutive attempts, with each sequence comprising a specified number of attempts and with successive sequences separated by a specified time interval.

m specifies the number of retries in each sequence. The maximum value is 255. (Any value less than 255 specifies the exact number of attempts; 255 specifies unlimited attempts.)

t specifies the interval, in seconds, between successive retry sequences. The maximum value is 255.

n specifies the number of times the retry sequence specified by *m* is to be performed. The maximum value is 255.

The maximum number of retries (retransmissions) equals $(n+1)\,m$.

If *m* is 0 (or is omitted), *t* must be 0 (or must be omitted). If *t* is 0 (or is omitted), *n* must be 0 (or must be omitted).

If you omit this operand entirely, the assumed values will be $m=2$ (for start-stop stations) or 7 (for BSC stations), $t=0$, $n=0$, if the type of station on the line is capable of retransmitting. That is, there will be one sequence of three retries. If the station is not capable of retransmission, *m* is assumed to be 0, that is, no retries at all are attempted.

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

Specifies whether or not the ring indicator mode of automatic answer operation is to be used for this line. This depends solely upon the type of modem (data set) that connects the line to the controller. Determine from the system designer whether the modem 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).

```
[SERVLIM=  count ]
```

Specifies the number of teleprocessing devices the network control program is to check each time the program performs service seeking. The maximum number you may specify in *count* is 255. If you omit this operand, a count of one is assumed. That is, each time the network control program performs service seeking, it checks only one device to determine if it requires servicing.

```
[SESSION= {count}]
          {1    }
```

Specifies the number of sessions the network control program is to attempt to maintain concurrently on a nonswitched multipoint line. The maximum number of sessions you may specify must not exceed the number of devices connected to the line. If this operand is omitted, no more than one session will be maintained on the line. (This operand is ignored if the LINE macro represents a nonswitched or switched point-to-point line, except for a nonswitched point-to-point line to which an IBM 1050 is connected.)

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

Specifies whether or not the controller is a tributary station on this line and if so, specifies its address.

If the controller is a tributary station, code TADDR=char, where *char* is a single EBCDIC character. Determine the address character from the system designer.

If the controller is not a tributary station, code TADDR=NONE or omit the operand.

```
[TRANS=( [inname][,outname] ) ]
```

Specifies the translation tables to be used for incoming and outgoing data. 'inname' specifies the name of a TRANSTBL macro which defines an input translation table; 'outname' specifies the name of a TRANSTBL macro which defines an output translation table. Every line must have one input and output translation table except for lines used for MTA.

If 'inname' is omitted, a standard input translation table for the device type is assumed; if 'outname' is omitted, a standard output translation table for the device type is assumed.

Note: *This operand should be omitted if TERM=MTA.*

[TRANSFR= {count / 7}]

Specifies a limit on the number of buffers that the network control program will obtain to receive message text from a station before the network control program transfers filled buffers to the host processor. If the network control program receives the specified number of buffers, it transfers them to the host processor as a sub-block, but continues to receive message text from the station until it receives an end-of-block character (or until the sub-block limit you have specified in the CUTOFF macro is reached).

Code TRANSFR=count, where *count* is from 1 to 255. If you wish to have a count of 7, you may omit the operand.

[YIELD= {YES / NO}]

Specifies whether or not the controller is to be considered the secondary station on a nonswitched point-to-point (contention) line. (The secondary station yields to the primary station when contention occurs.)

Code YIELD=YES (or omit the operand) if you wish the controller to yield to the remote (primary) station when contention occurs. Code YIELD=NO if you wish the controller to be the primary station.

If you code YIELD=YES (or omit the operand), the remote station should be arranged to act as a primary station; if you code YIELD=NO, the remote station should be arranged to act as a secondary station.

In addition to the preceding operands, most operands of the CLUSTER, TERMINAL, and COMP macros can be specified in the LINE macro (or the GROUP macro) instead of the individual macros mentioned. Figure 6 shows which of the lower-level operands you may specify in the LINE macro.

CLUSTER Macro Instruction

The CLUSTER macro represents a station of the "clustered" type (IBM 3270, 3275, or 2972) and specifies:

- The type of station.
- Whether the station has the Batch Message Input feature (2972 only).
- The general polling characters of the station, if required.
- The initial status of the station when the network control program is loaded into the controller.
- The unit of data transfer the network control program will consider a record when received from the station.
- The block handler set, if any, associated with the station, and the points of execution of block handlers within the set.

Name	Operation	Operands
symbol	CLUSTER	[BHEXEC= {([PT1][,PT2][,PT3]) ALL}] [,BHSET= {NONE DYNAMIC setname[,EXEC={YES NO}]}] [,CUTYPE={3271 3275 2972}] [,FEATURE= {BATCH NOBATCH}] [,GPOLL=chars] [,ISTATUS= {INACTIVE (ACTIVE[, {WAIT INVITE[,{modifier B}]}])}] [,RECORD={BLOCK MESSAGE XMISSION}]

symbol

Is any valid symbol. It provides a name for the station and is required.

[BHEXEC={([PT1][,PT2][,PT3])
ALL}]

Specifies which block handlers in the set specified by BHSET are to be executed.

[PT1]

Specifies that the network control program is to execute the PT1 block handler. It will execute this block handler upon receiving a request from the host processor but before determining whether the line is available to contact the station.

[PT2]

Specifies that the network control program is to execute the PT2 block handler. It will execute this block handler upon receiving a request from the host processor *and* after determining that the line is available.

[PT3]

Specifies that the network control program is to execute the PT3 block handler. It will execute this block handler after receiving a block, message, or transmission (as determined by the RECORD operand) from the station.

ALL

Specifies that each block handler in the set is to be executed at the appropriate time.

If this operand is omitted, and BHSET is specified, ALL is assumed.

Note 1: *For a description of how a block handler set is defined, see the section, Block Handling Definition (BH) Macro Instructions.*

Note 2: *Code this operand only if you specify* setname *in the BHSET operand. BHEXEC will be ignored if BHSET specifies NONE or DYNAMIC.*

Note 3: *At least one execution point must be specified if the BHSET operand specifies* setname. *If BHSET specifies NONE or DYNAMIC, do not specify an execution point.*

```
[BHSET= { NONE                                    }
        { DYNAMIC                                 }
        { setname[,EXEC={YES} ] }
        {              {NO}   }
```

Specifies the name of a set of block handlers to be associated with this station.

NONE

Specifies that no block handler set is to be assigned to this station, and no dynamic assignment of block handler sets can be done.

DYNAMIC
Specifies that no block handler set is to be initially assigned to this station, but a block handler set may be assigned dynamically.

setname

Specifies the block handler set for this station; it must be the *setname* appearing on a BHSET macro.

[EXEC= {YES / NO}]

Specifies whether or not the block handler set is to be initially executable. Code EXEC=YES (or omit the operand) if the block handler set is to be initially executable. Code EXEC=NO if you do not wish the block handler set to be initially executable. If you code EXEC=NO, the block handler set must be activated by a request from the host processor.

[CUTYPE= {3271 / 3275 / 2972}]

Specifies whether the control unit of the station is a 3271, a 3275, or a 2972.

IBM 2972: Following each CLUSTER macro that represents an IBM 2972 General Banking Terminal System, you must code a sequence of TERMINAL macros, one to represent each 2980 attached to the 2972. Specify in the ADDR operand the addressing character assigned to that 2980. (Determine the character from the system designer.) Since 2980s cannot be individually polled, the POLL operand is invalid.

IBM 3270: Following each CLUSTER macro that represents an IBM 3271 (CUTYPE=3271) you must code a sequence of TERMINAL macros, one for each 3277, 3284, and 3286 attached to the 3271. In the POLL and ADDR operands of each TERMINAL macro, specify the individual polling and addressing characters assigned to the terminal. (Determine the character from the system designer.)

[FEATURE= {BATCH / NOBATCH}] (IBM 2972 only)

Specifies whether or not the station has the batched message input feature (2972 only).

[GPOLL=chars]

Specifies that the general polling procedure is to be used for the station represented by the CLUSTER macro, and specifies the general polling characters assigned to the control unit of the station. If you omit this operand, terminals must be individually polled.

```
[ISTATUS= {INACTIVE                                                    }  ]
          {(ACTIVE[,{WAIT                                  }  ])       }
          {         {INVITE[,  {modifier}  ]               }           }
          {                    {B        }                 }           }
```

Specifies whether or not the station is to be logically active initially (i.e., available for data transmission as soon as the network control program begins executing after being loaded into the controller).

INACTIVE

Specifies that the station is not logically active initially. The network control program rejects any requests it receives for this station until the host processor sends a request, via the dynamic control facility, to change the status to active.

```
(ACTIVE[,{WAIT                                  }  ])
         {   INVITE[,{modifier}  ]              }
         {           {B        }                }
```

ACTIVE

Specifies that the station is logically active initially.

WAIT

Specifies that the network control program is to wait for an Invite request from the host processor before inviting the station to transmit.

INVITE

Specifies that the network control program is to invite the station to transmit immediately after network control program execution begins.

```
[{modifier}]
 {B        }
```

Specifies the type of Invite operation the network control program is to perform.

modifier

Specifies one of the following modifiers for the Invite operation.

Modifier	*Meaning*
B	Block
M	Message
T	Transmission
D	Transmission with Disconnect
A	With Automatic Restart

If this suboperand is omitted, B is assumed.

```
[RECORD= {BLOCK   }]
         {MESSAGE }
         {XMISSION}
```

Specifies what unit of data transfer (record) the network control program is to receive from the station during one Invite or Read operation.

If you code RECORD=BLOCK, the network control program will receive one *block* and pass it to the host processor. A data block is a sequence of characters ending with an EOB or EOT character (for start-stop stations) or ETB, ETX, or EOT (for BSC stations).

If you code RECORD=MESSAGE, the network control program will receive data blocks (ended by EOB or ETB) in succession until it has received an entire physical *message*. A message from a start-stop station ends with an EOT character; one from a BSC station ends with an ETX or EOT character.

As it receives individual blocks from the station, the network control program passes them to the host processor (i.e., the network control program does not accumulate the entire message before sending to the host processor).

If you code RECORD=XMISSION, the network control program will receive data blocks (ended by EOB or ETB) in succession until it has received an entire *transmission*. A transmission from either a start-stop or a BSC station ends with an EOT character.

As it receives individual blocks from the station, the network control program passes them to the host processor (i.e., the network control program does not accumulate the entire transmission before sending to the host processor).

Notice that specifying MESSAGE or XMISSION in the RECORD operand will have the same effect for Read and Invite operations with a start-stop station: the network control program will continue receiving from the station until it detects an EOT character. This is because in start-stop transmission, no distinction exists between messages and transmissions; the EOT character signifies the end of transmission or end of message.

Note: *The unit of data transfer you specify in the RECORD operand governs the Invite or Read operation only for those requests for which the 'null' modifier is specified. The purpose of the 'null' modifier is to specify that the unit you have designated in the RECORD operand is to govern the operation. If, on the other hand, the block, message or transmission modifier is set in the control information accompanying the request, the unit of data transfer specified by the modifier governs that particular request.*

In addition to the preceding operands, most operands of the TERMINAL macro can be specified in the CLUSTER macro (for cluster-type stations) or in the LINE or GROUP macro, instead of the TERMINAL macro. Figure 6 shows which of the TERMINAL macro operands you may specify in the CLUSTER macro.

TERMINAL Macro Instruction

The TERMINAL macro represents a station in the teleprocessing subsystem and specifies:

- The type of station
- The features with which the station is equipped
- The polling and addressing characters by which the network control program will contact the station (multipoint line control)
- The telephone number by which the network control program can reach the station (switched line)
- The length of the controller ID sequence the network control program will send to the station when contact is established
- The block handler set, if any, associated with the station, and the times of execution of block handlers within the set
- Certain procedural options the network control program is to use when communicating with the station.

Name	Operation	Operands
symbol	TERMINAL	TERM=type [,ADDR=chars] [,ATTN= {ENABLED / DISABLED}] [,BFRDLAY= {count / 0}] [,BHEXEC={([PT1][,PT2][,PT3]) / ALL}] [,BHSET= {NONE / DYNAMIC / setname[,EXEC={YES / NO}]}] [,CRITSIT= {NO / YES}] [,CTERM= {NO / YES}] [,CUIDLEN= {n / 0}] [,DIALNO=([chars][,count]) [,DIALSET= {dialset name / NONE}] [,ENDTRNS= {EOT / EOB}] [,FEATURE=([{ACR / NOACR}] [,{ATTN / NOATTN}] [,{BREAK / NOBREAK}] [,{CHECK / NOCHECK}] [,{CONV / NOCONV}] [,{DUAL / NODUAL}] [,{SCTL / NOSCTL}] [,{TOSUPPR / NOTOSUP}] [,{XCTL / NOXCTL}] [,ISTATUS= {INACTIVE / (ACTIVE[{WAIT / INVITE[,{modifier / B}]}])}] [,POLL=chars] [,RECORD={BLOCK / MESSAGE / XMISSION}] [,XMITLIM= {COUNT / NO}]

symbol

Is any valid symbol. It provides a name for the station and is required.

TERM=type

Specifies the type of station represented by this TERMINAL macro. It must be one of the types listed in Figure 8.

This operand is required.

Type	Description
1030	IBM 1030 Data Collection System
1050	IBM 1050 Data Communication System
1060	IBM 1060 Data Communication System
1130	IBM 1130 Computing System
1800	IBM 1800 Data Acquisition and Control System
2020	IBM System/360 Model 20
2025	IBM System/360 Model 25
2701	IBM 2701 Data Adapter Unit
2703	IBM 2703 Transmission Control
2715	IBM 2715 Transmission Control Unit Model 2
2740-1	IBM 2740 Model 1 Communications Terminal
2740-2	IBM 2740 Model 2 Communications Terminal
2741	IBM 2741 Communications Terminal
2760	IBM 2760 Optical Image Unit
2770	IBM 2770 Data Communications System
2780	IBM 2780 Data Transmission Terminal
2980	IBM 2980 Models 1 and 4 Teller Station[1]
2980	IBM 2980 Model 2 Administrative Station[1]
3275	IBM 3275 Display Station
3277	IBM 3277 Display Station[2]
3284	IBM 3284 Printer[2]
3286	IBM 3286 Printer[2]
3735	IBM 3735 Programmable Buffered Terminal
3705	IBM 3705 Communications Controller
MTA	Multiple Terminal Access (IBM 1050, 2740, 2741)
3135	IBM System/370 Model 135
SYS3	IBM System/3
2740-1	IBM System/7 (supported as a 2740 Model 1)
83B3	
115A	
TWX	
WTTY	

[1]Attached to an IBM 2972 Model 8. Specify TERM=2980 only in a TERMINAL macro that follows a CLUSTER macro in which CUTYPE=2972 is coded.

[2]Valid only in a TERMINAL macro that follows a CLUSTER macro in which CUTYPE=3271 is coded.

Figure 8. Values for TERM operand of TERMINAL Macro Instruction

[ADDR=chars]

Specifies (1) the EBCDIC characters with which the network control program is to address the station, if the station is on a multipoint line; (2) the EBCDIC component-selection escape sequence for one of the output components, if the station is on a point-to-point line; or (3) the EBCDIC characters with which the network control program is to address the terminal represented by this TERMINAL macro, if the station is a clustered-type station (i.e., IBM 2972, 3271, or 3275).

The addressing character assigned to the station should be determined from the system designer.

[ATTN= { ENABLED / DISABLED }] (IBM 1050, 2741; TWX)

Specifies whether the Attention feature of the network control program is to be initially enabled or disabled. When the feature is enabled, an Attention signal received from the terminal causes the network control program to stop sending to the terminal. The Attention signal has the same effect on the network control program as a Reset Immediate command.

ATTN=ENABLED is valid only if FEATURE=ATTN is coded in this TERMINAL macro.

[BFRDLAY={ count / 0 }]

Specifies the delay, in seconds, between successive transmissions to the device represented by this TERMINAL macro. This operand is valid only for buffered devices, as follows:

IBM 2740 Model 2 with the buffered receive feature
IBM 2770
IBM 2980
IBM 3284
IBM 3286

[BHEXEC={ ([PT1][,PT2][,PT3]) / ALL }]

Specifies which block handlers in the set specified by BHSET are to be executed.

[PT1]

Specifies that the network control program is to execute the PT1 block handler. It will execute this block handler upon receiving a contact or write request from the host processor but before it determines whether the line is available to contact the station.

[PT2]

Specifies that the network control program is to execute the PT2 block handler. It will execute this block handler upon receiving a contact or write request from the host processor or after message data has been received from the line. The network control program will execute the block handler while the line is available for sending to or receiving from the station.

[PT3]

Specifies that the network control program is to execute the PT3 block handler. It will execute this block handler after receiving a block, message, or transmission (as determined by the RECORD operand) from the station.

ALL

Specifies that each block handler in the set is to be executed at the appropriate time.

If this operand is omitted, and BHSET is specified, ALL is assumed.

Note: *At least one execution point must be specified.*

Note 1: *For a description of how a block handler set is defined, see the section,* Block Handler Definition Macro Instructions .

Note 2: *Code this operand only if you specify* setname *in the BHSET operand. BHEXEC will be ignored if you have coded BHSET=NONE or BHSET=DYNAMIC, or if you have omitted the BHSET operand.*

```
[BHSET={ NONE                                  }  ]
       { DYNAMIC                               }
       { setname[,EXEC={YES}]                  }
                       {NO }
```

Specifies the name of a set of block handlers to be associated with this station.

NONE

Specifies that no block handler set is to be assigned to this station and no dynamic assignment of block handler sets can be done.

DYNAMIC

Specifies that no block handler set is to be initially assigned to this station but a block handler set may be assigned dynamically.

setname

Specifies the block handler set for this station; it must be the *setname* naming a BHSET macro.

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

Specifies whether or not the block handler set is to be initially executable. Code EXEC=YES (or omit the operand) if the block handler set is to be initially executable. Code EXEC=NO if you do not wish the block handler set to be initially executable. If you code EXEC=NO, the block handler set must be activated by a request from the host processor.

[CRITSIT= {NO / YES}]

Specifies whether or not this station, if logically active, is to be notified when the network control program is about to close down the teleprocessing network because a critical situation, such as channel failure, has occurred.

Code CRITSIT=YES if you wish this station, if logically active, to be notified. Code CRITSIT=NO (or omit the operand) if you do not wish the station to be notified.

If you code CRITSIT=YES, also specify, in the CSMSG operand of the BUILD macro, the text of the message. If the CSMSG operand is omitted, all CRITSIT operands, if coded, are ignored.

Note: *The network control program sends the notification message to the output component represented by the TERMINAL macro, not to any output component represented by a COMP macro.*

[CTERM= {NO / YES}] (switched lines only)

Specifies whether or not this TERMINAL is to represent a "logical connection" station. This is not an actual station in the teleprocessing network; rather, the network control program uses the control fields generated by this TERMINAL macro to hold control information about any station that calls the controller over a switched line. The network control program uses these control fields successively for various stations that call over the line represented by the LINE macro preceding this TERMINAL macro.

One TERMINAL macro in which CTERM=YES is coded is required following each LINE macro in which CALL=IN or CALL=INOUT is coded, or in which the CALL operand is omitted. It is not required, and should be omitted, for a LINE macro in which CALL=OUT is coded.

[CUIDLEN= {n / 0}] (switched BSC lines only)

Specifies that the network control program is to send the ID sequence of the controller to the BSC station and provides the length of the controller ID sequence. The first *n* characters of the ID specified by the CUID operand on the BUILD macro will be transmitted. The maximum value for *n* is 15.

If you omit this operand or code it as CUIDLEN=0, the network control program will not send the ID of the controller to the station. (The network control program may, however, verify ID sequences it receives from stations.)

[DIALNO=([chars][,count])]

[chars]

Specifies the telephone number of the station. *chars* represents the dial digits the network control program will use to contact the station over a switched line. You may specify this parameter either if the station is reached solely by a switched connection or if the switched connection is an alternate to the primary nonswitched point-to-point line.

[count]

Specifies the number of bytes to be reserved to hold the telephone number.

Use these two suboperands as follows:

(1) If the station will always be called using the dial digits you specify in this operand, code *chars* and omit *count*.
(2) If the host processor will update the dial digits using the dynamic control facility, (DIAL must be included in the options specified by the SYSCNTRL macro), code *chars*. Also code *count* if the number of replacement dial digits may exceed the number of digits originally specified by the *chars* suboperand. Otherwise, you may omit the *count* suboperand.
(3) If the host processor is to supply the station's telephone number with each contact request, code the size of the longest number in the *count* suboperand; you must omit the *chars* suboperand.

Note: *The dash character (-) may be inserted one or more times in the telephone number to indicate a dialing pause. When the network control program encounters the dash character while dialing the station, the program pauses for one second before dialing the next digit. You may specify one or more dash characters at any point in the telephone number. (This pause is useful when time must be allowed to elapse for receiving a secondary dial tone.) Include the dash characters in the number of dial digits in the count suboperand.*

[DIALSET= {dialset name / NONE}] (switched line only)

Specifies the name of the primary (or only) dial set to be associated with the station represented by this TERMINAL macro.

Each time the network control program receives a call-out request for the station represented by this TERMINAL macro, it attempts to call that station over one of the switched lines in the specified dial set.

Rules for use of this operand are as follows:

- If you wish the network control program to call the station via the dial set containing the line whose LINE macro precedes this TERMINAL macro, omit the DIALSET operand in this macro. The DIALSET operand of the LINE macro specifies the dial set used.
- If you wish the network control program to call the station via a dial set different from the dial set containing the line whose LINE macro precedes this TERMINAL macro, specify the desired dial set in the DIALSET operand of this TERMINAL macro.
- If no dial set needs to be associated with this station because the network

control program will not call the station, code DIALSET=NONE. This choice is appropriate when the station is on a nonswitched point-to-point or multipoint line (DIAL=NO [or operand omitted] in the GROUP macro). It is also appropriate when the station can call the communications controller but the network control program cannot call the station (that is, the LINE macro preceding this TERMINAL macro specifies CALL=IN).

```
[ENDTRNS= {EOT}]
          {EOB}
```

Specifies whether the network control program is to consider a transmission terminated by an end-of-transmission character (EOT) or by an end-of-block character (EOB).

Note: *This operand is valid only for start-stop terminals that can transmit an EOB EOT ending sequence.*

```
[FEATURE=...]
```

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

```
[ {ACR  }]          (IBM 1050 only)
  {NOACR}
```

Specifies whether or not the station is equipped with the Accelerated Carrier Return feature.

```
[ {ATTN  } ]        (IBM 1050, 2741 only)
  {NOATTN}
```

Specifies whether or not the station is equipped with the Receive Interrupt (IBM 1050) or Interrupt (IBM 2741) feature; that is whether the station can interrupt the controller while the controller is sending data to the station.

Note: *ATTN is valid for TWX.*

```
[ {BREAK  } ]       (IBM 1050, 2741 only)
  {NOBREAK}
```

Specifies whether or not the station is equipped with the Transmit Interrupt feature; that is, whether the controller can interrupt the station while the station is transmitting to the controller.

Note: *Break is valid for TWX.*

```
[ {CHECK  }]        (IBM 2740 only)
  {NOCHECK}
```

Specifies whether or not the station is equipped with the Checking feature.

```
[ {CONV   } ]     (BSC stations only)
  {NOCONV }
```

Specifies whether or not the station is equipped with the Conversational Mode feature.

```
[ {DUAL   } ]
  {NODUAL }
```

Specifies whether or not the station is equipped with the Dual Communications Interface feature. (The DIALNO operand must also be specified, if you code FEATURE=DUAL.)

```
[ {SCTL   } ]     (IBM 2740 only)
  {NOSCTL }
```

Specifies whether or not the station is equipped with the Station Control feature. (This feature is optional for the IBM 2740 Model 1, standard for the Model 2.)

```
[ {TOSUPPR} ]
  {NOTOSUP}
```

Specifies whether or not the station is equipped with the Text Timeout Suppression feature.

```
[ {XCTL   } ]          (IBM 2740 Model 1 only)
  {NOXCTL }
```

Specifies whether or not the station is equipped with the Transmit Control feature.

```
[ISTATUS= {INACTIVE                                                  }
          {(ACTIVE[,  {WAIT                                  } ])  }  ]
          {           {INVITE[, {modifier} ]                 }     }
          {                     {B       }                       }
```

Specifies whether or not the network control program is to consider the station operational initially (i.e., available for data transmission as soon as the network control program begins executing after being loaded into the controller).

INACTIVE

Specifies that the station is initially not operational (i.e., inactive). The network control program will reject any requests it receives for this station until the host processor changes the status to active, via the dynamic control facility.

ACTIVE

Specifies that the station is initially operational.

WAIT

Specifies that the network control program is to wait for a request from the host processor before communicating with the station.

INVITE

Specifies that the network control program is to initially invite the station to transmit.

```
[   {modifier}  ]
    {B        }
```

Specifies the type of Invite the network control program is to issue.

modifier

Specifies one of the following modifiers for the Invite command:

Modifier	*Meaning*
B	Block
M	Message
T	Transmission
D	Transmission with Disconnect
A	With Auto Restart

If this parameter is omitted, B is assumed.

[POLL=chars]

Specifies the EBCDIC characters to be used in polling this station.

Note: *This operand is invalid if the TERMINAL macro represents an IBM 2980, as this type of terminal cannot be individually polled.*

```
[RECORD={BLOCK  }  ]
        {MESSAGE}
        {XMISSION}
```

Specifies what unit of data transfer (record) the network control program is to receive from the station during one Invite or Read operation.

If you code RECORD=BLOCK, the network control program will receive one *block* and pass it to the host processor. A data block is a sequence of characters ending with an EOB or EOT character (for start-stop stations) or ETB, ETX, or EOT (for BSC stations).

If you code RECORD=MESSAGE, the network control program will receive data blocks (ended by EOB or ETB) in succession until it has received an entire *message*. A message from a start-stop station ends with an EOT character; one from a BSC station ends with an ETX or EOT character.

As it receives individual blocks from the station, the network control program passes them to the host processor (i.e., the network control program does not accumulate the entire message before sending to the host processor).

If you code RECORD=XMISSION, the network control program will receive data blocks (ended by EOB or ETB) in succession until it has received an entire *transmission*. A transmission from either a start-stop or a BSC station ends with an EOT character.

As it receives individual blocks from the station, the network control program passes them to the host processor (i.e., the network control program does not accumulate the entire transmission before sending to the host processor).

Notice that specifying MESSAGE or XMISSION in the RECORD operand will have the same effect for Read and Invite operations for a start-stop station: the network control program will continue receiving from the station until it detects an EOT character. This is the case because in start-stop transmission there is no distinction between messages and transmissions; the EOT character signifies the end of transmission or end of message.

Note: *The unit of data transfer you specify in the RECORD operand governs the Invite or Read operation only for those requests for which the 'null' modifier is specified. The purpose of the 'null' modifier is to specify that the unit you have designated in the RECORD operand is to govern the operation. If, on the other hand, the block, message or transmission modifier is specified for any request, the unit of data transfer indicated by the modifier governs that particular request.*

```
[XMITLIM= {count} ]
          {NO   }
```

Specifies the maximum number, if any, of transmissions (ended by EOT unless you have specified ENDTRNS=EOB) the network control program will receive from or send to this station or both. If this limit is reached before the host processor explicitly requests that the network control program disconnect the station from the controller, the network control program will automatically disconnect.

The maximum value you may specify is 255.

XMITLIM=NO means that the network control program will send to, or receive from, the station indefinitely.

In addition to the previous operands, most operands of the COMP macro can be specified in the TERMINAL (or LINE or GROUP) macro instead of in the COMP macro. Figure 6 shows which of the COMP macros you may specify in the TERMINAL macro.

COMP Macro Instruction

The COMP macro represents one input component, or one output component, or both, of a station in the teleprocessing subsystem. COMP also specifies:

- The polling and addressing characters by which the network control program will contact the components.
- The block handler set, if any, associated with the components, and the times of execution of block handlers within the set.
- Certain procedural options the network control program is to use when communicating with the component.

All macros defining components of a station must be grouped together immediately following the TERMINAL macro representing the station to which the components are attached.

COMP macros are required only for components the network control program must individually poll or address, using the specific polling or addressing characters assigned to the component. If the teleprocessing access method has no need to send to the network control program requests naming specific components, you should avoid using COMP macros. Each COMP macro generates control tables that occupy additional storage space in the communications controller.

Name	Operation	Operands
symbol	COMP	[ADDR=chars] [,BHEXEC={([PT1][,PT2][,PT3]) ALL}] [,BHSET={NONE DYNAMIC setname[,EXEC={YES NO}]}] [,ENDTRNS={EOT EOB}] [,ISTATUS={INACTIVE (ACTIVE[,{WAIT INVITE[,{modifier B}]}])}] [,POLL=chars] [,RECORD={BLOCK MESSAGE XMISSION}] [,XMITLIM={count NO}]

symbol

Is any valid symbol. It provides a name for the component and is required.

The operands shown above have meanings equivalent to, and are to be used similarly to, the correspondingly named operands of the TERMINAL macro. See the TERMINAL macro for descriptions of the operands.

Block Handler Definition Macro Instructions

With these macro instructions you specify what processing operations the network control program is to perform on blocks of message data received from the host processor or from stations. Each function is performed by a routine called a block handling routine. Several block handling routines may be grouped into a block handler; the routines process the data in the same sequence as the block handling routine macros appear within the block handler.

You may establish a set of block handlers for each station, with each set including any of the block handling routines appropriate for the station. You may also specify separate block handler sets for individual components of a station.

Block Handler Delimiter Macro Instructions

The two macros STARTBH and ENDBH define the beginning and end of a block handler. Only one STARTBH and ENDBH may appear in a block handler.

STARTBH Macro Instruction

The STARTBH macro:

- Establishes the beginning of a block handler.
- Provides a name for the block handler.

Name	Operation	Operands
bhname	STARTBH	[BHEXEC= {PT1 / PT2 / PT3}]

bhname

Is any valid symbol. It provides a name for the block handler and is required. (This name is referred to by the BHSET macro.)

[BHEXEC={PT1 / PT2 / PT3}]

Specifies the point at which this block handler will be executed. When the block handler is to process outgoing message data (data being *sent to* a station):

If you wish the data to be processed immediately upon being received from the host processor, *before* the network control program is ready to send it to the station, code BHEXEC=PT1. *(Ready to send it* means the network control program has contacted the station and is prepared to send the data block at once.)

If you wish the block handler *not* to process the data immediately but to wait until the network control program is ready to send the data block to the station, code BHEXEC=PT2.

The block handling routines you can include in a block handler that processes outgoing data are Date and Time, or any user-provided block handling routines. Therefore, a STARTBH macro that specifies BHEXEC=PT1 or PT2 may be followed by DATETIME, or UBHR macros.

If the block handler is to process incoming data as well as outgoing data (this is possible only if you code BHEXEC=PT2), you may also include the EDIT macro. However, the block handling routine invoked by this macro will process only the *incoming* data.

Block handlers process outgoing data only if the data transfer command is not in error.

When the block handler is to process *incoming* message data (data received from a station):

If you wish the data to be processed immediately upon being received from the station, *before* the network control program continues the session or breaks the logical connection with the station, code BHEXEC=PT2.

If you wish to allow the network control program to continue with the session or to break the logical connection, before the received message data is processed, code BHEXEC=PT3.

The block handling routines you can include in a block handler that processes incoming data are Date and Time, Edit, or any user-provided block handling routines. Therefore, a STARTBH macro that specifies BHEXEC=PT2 or PT3 may be followed by an DATETIME, EDIT, or UBHR macro.

Except for the Date and Time routine and any user block handling routines, block handlers process incoming data only if the data block was correctly received (that is, a data check error did not occur).

The ACCESS operand of the UBHR macro determines whether the routine processes only error-free blocks, or only error-containing blocks.

If you omit the BHEXEC operand, the block handler will be executed as if BHEXEC=PT1 were specified.

ENDBH Macro Instruction

The ENDBH macro:

- Specifies the end of a block handler.
- Must be the last macro of each block handler.

Name	Operation	Operands
[symbol]	ENDBH	

[symbol]

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

This macro has no operands.

Block Handling Function Macro Instructions

Function macros specify the individual block-handling routines to be included in the block handler. Each routine performs a particular function on the data being processed. The order of the function macros in the block handler determines the order in which the functions will be performed.

DATETIME Macro Instruction

The DATETIME macro specifies whether the network control program is to insert the date, the time of day, or both, in a data block. The macro also specifies whether to insert this information in only the first block of a message or in all blocks. It also specifies one of four formats in which the date is to appear.

The network control program inserts the date and/or time as the first data in a block. The date and time immediately precede the first text character in a block. The date precedes the time.

Name	Operation	Operands
[symbol]	DATETIME	[DATE={ YES / NO }] [,DATEFMT={ YY.DDD / MM/DD/YY / YY/MM/DD / DD/MM/YY }] [,INSERT={ FIRST / ALL }] [,PT2EXEC={ BEFORE / AFTER }] [,TIME={ YES / NO }]

[DATETIME] [ENDBH]

[symbol]

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

[DATE={ YES / NO }]

Specifies whether or not the current date is to be inserted.

[DATEFMT={ YY.DDD / MM/DD/YY / YY/MM/DD / DD/MM/YY }]

Specifies the format for the date.

Code DATEFMT=YY.DDD or omit the operand if you wish the date to appear as the year followed by the day of year; for example, 73.294 (October 21, 1973).

Code DATEFMT=MM/DD/YY if you wish the date to appear in the month/day/year format; for example, 10/21/73.

Code DATEFMT=YY/MM/DD if you wish the date to appear in the year/month/day format; for example, 73/10/21.

Code DATEFMT=DD/MM/YY if you wish the date to appear in the day/month/year format; for example, 21/10/73.

```
[INSERT={FIRST}]
        {ALL  }
```

Specifies whether the date and/or time is to be inserted in the first block of each message or in all blocks.

```
[PT2EXEC= {BEFORE}]
          {AFTER }
```

Specifies, for a routine executed at PT2, whether the routine is to insert the date and/or time *before* the I/O Operation (PT2EXEC=BEFORE) or after the I/O Operation (PT2EXEC=AFTER).

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

Specifies whether or not the current time is to be inserted. The time is always in the format hh.mm.ss, using the continental (24-hour) form. For example, 07.42.18 represents 7:42:18 a.m.; 19.42.18 represents 7:42:18 p.m.

Note 1: *Do not code both DATE=NO and TIME=NO.*

Note 2: *If you specify both EDIT and DATETIME in the same block handler, the EDIT macro must precede DATETIME.*

EDIT Macro Instruction

The EDIT macro causes the network control program to edit data originally entered from a keyboard such that erroneously entered characters, when followed by a text canceling character, are deleted from the message data. Example: the network control program will correct PENNSLYV///YLVANIA to PENNSYLVANIA, if the / character the keyboard operator used as the text canceling ("backspace") character is specified in the EDIT macro.

Name	Operation	Operands
[symbol]	EDIT	[BKSP= {char / 16}]

[symbol]

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

[BKSP= {char / 16}]

Specifies the character acting as the text canceling (backspace) character. *Char* is the hexadecimal representation of the text canceling character EBCDIC without the framing characters X" (e.g., 4C). If this operand is omitted, 16 (the hexadecimal representation of the EBCDIC backspace (BS) character) is assumed to be the text canceling character.

Note 1: *The EDIT macro may only be specified in a block handler that processes data received from a station. It cannot be specified following a STARTBH macro in which BHEXEC=PT1 is coded (a block handler executed at point 1 acts only upon data being sent to a station).*

Note 2: *If you specify both EDIT and DATETIME in the same block handler, the EDIT macro must precede DATETIME.*

UBHR Macro Instruction

The UBHR macro allows you to include a user-written block handling routine in a block handler. It specifies the name of the module and its entry point, and specifies under what conditions it is to be executed.

Note: *Use of the UBHR macro forces storage boundary alignment to the next 2K boundary because of the storage-protect feature of the communications controller. Up to 2K bytes of storage may therefore be unused when the network control program is loaded into the controller.*

Name	Operation	Operands
[symbol]	UBHR	NAME=member name [,ACCESS={GOOD | ERROR}] [,COMMAND={READ | INVITE | BOTH}] [,ENTRY=entry point name] [,PT2EXEC={BEFORE | AFTER}]

[symbol]

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

NAME=member name

Specifies the name of a user-written module. The module named must be contained in the data set specified by the USERLIB operand of the BUILD macro.

[ACCESS={GOOD | ERROR}]

Specifies whether the user block handling routine is to process only good (error-free) blocks, or blocks containing errors.

ACCESS=ERROR is valid only if the UBHR macro follows a STARTBH macro in which BHEXEC=PT3 or BHEXEC=PT2 is specified. In the latter case, PT2EXEC=AFTER must also be coded in the UBHR macro. If this UBHR macro follows a STARTBH macro that specifies BHEXEC=PT1, this operand is invalid and must be omitted.

[COMMAND={READ | INVITE | BOTH}]

Specifies whether the user-written routine is to process data received in response to a Read request, an Invite request, or both. This applies only for incoming data, that is, when the UBHR macro follows a STARTBH macro that specifies BHEXEC=PT2 (in which case PT2EXEC=AFTER must be coded) or that specifies BHEXEC=PT3.

If this UBHR macro follows a STARTBH macro that specifies BHEXEC=PT1, this operand is invalid and must be omitted.

[ENTRY=entry point name]

Specifies the name of the entry point within the user-written module named by the NAME operand. If you omit the *ENTRY* operand, the entry point name is assumed to be the same as the module name.

[PT2EXEC={BEFORE / AFTER}]

Specifies whether the user routine is to process outgoing message data *before* the I/O operation, (PT2EXEC=BEFORE); or to process incoming message data *after* the I/O operation (PT2EXEC=AFTER).

This operand is valid only in a UBHR macro that follows a STARTBH macro that specifies BHEXEC=PT2.

Block Handler Set Macro Instruction (BHSET)

BHSET defines a set of block handlers which may be statically or dynamically assigned to a device.

Name	Operation	Operands
setname	BHSET	[PT1=bhname] [,PT2=bhname] [,PT3=bhname]

setname

Is any valid symbol. It provides a name for the set of block handlers and is required. (It is referred to by the BHSET operand of the CLUSTER, TERMINAL, and COMP macros.

[PT1=bhname]

Specifies the name of the block handler to be executed when a request has been received from the host processor for the device but before it has been determined that the line is available.

[PT2=bhname]

Specifies the name of the block handler to be executed when a request has been received from the host processor for the device and after the line has been found to be available.

[PT3=bhname]

Specifies the name of the block handler to be executed at PT3. It will be executed when an input operation on a communication line completes, and after the line is released for use with another station.

Generation Delimiter Macro Instruction (GENEND)

The GENEND macro indicates the end of the network control program generation input deck. It must be the last network control program generation macro instruction coded.

Name	Operation	Operands
[symbol]	GENEND	[SCANCTL=(limit1, limit2, limit3, limit4, asmask)]

[symbol]

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

[SCANCTL=(limit1,limit2,limit3,limit4,asmask)]

Specifies the scan limits for the Type 2 communication scanners installed in the controller and specifies the address substitution mask, if used.

limit1...limit4

Specifies the scan limits for the scanners. Each limit can be from 0 to 4; these values have the meanings shown below. Limit1 specifies the scan limit for the first scanner position, limit2 for the second, etc. If a scanner position is vacant, that is, if no scanner is installed at that position, code a comma for the corresponding limit. If a scanner is installed but no value is coded, a scan limit of 0 is assumed.

The scan limits have the following meanings:

Scan Limit	*Addresses Scanned*	*Addresses Not Scanned*	*Maximum Line Speed*
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	50,000 bps
2	020-04F 0A0-0CF	050-05F 0D0-0FF	9,600 bps

	120-14F	150-17F	
	1A0-1CF	1D0-1FF	
3	020-02F	030-05F	19,200 bps
	0A0-0AF	0B0-0FF	
	120-12F	130-17F	
	1A0-1AF	1B0-1FF	

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. Addresses A and B in all LIB positions are disabled.
2	0	No address substituion; 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.

The address substitution mask governs address substitution in all installed Type 2 communication scanners.

Note; *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.*

Chapter 4: The Network Control Program Generation Procedure

Note: *The information in this chapter is preliminary. Detailed information on generating the network control program will be provided in an updated version of this publication.*

The network control program generation procedure is a two-stage process. This process consists of a series of jobs executed under control of the operating system either in the host processor or in a different central processing unit. Stage 1 is an assembly job using the 3705 Assembler or OS assembler to prepare, from the network control program generation macros, a sequential data set for input to stage 2. This data set may be placed on cards, tape, or direct-access devices. The stage 1 output (stage 2 input) data set contains (1) data constants and stage 2 macros that generate control tables and conditionally-assembled modules, (2) job control statements, and (3) linkage editor control statements.

Stage 2 of the generation procedure assembles control tables, then link-edits the selected network control program object modules, control tables, and any user-written translation tables and block handling routines. The result is a network control program load module, ready to be loaded into the communications controller from the host processor.

Stage 2 also produces a *resource resolution table* load module. This load module contains information the teleprocessing access method needs in order to specify the proper resource for requests.

Operator intervention is required between the two stages of generation. Diagnostic messages produced at the end of stage 1 indicate any errors that have occurred. If these errors are serious, no job stream or partial job stream is produced. The source statements must be corrected and stage 1 must be re-executed. If no serious assembly errors occur, the operator initiates the second stage, specifying as input the Stage 1 output.

Figures 9 and 10 show the content of the stage 1 input job stream and stage 1 output (stage 2 input) job stream.

```
//STAGE1 JOB  MSGLEVEL=1
//STEP1  EXEC PGM=IFKASM
       .
       .
   JCL for Assembler
       .
       .
//SYSIN  DD   *
       .
       .
   Network Control Program Generation Macro
   Instruction Statements
       .
       .
/*
```

Figure 9. Stage 1 Input Job Stream

```
//STAGE2 JOB  MSGLEVEL=1     Stage2 has multiple assemblies
                             as required
//STEP1  EXEC PGM=IFKASM
       .
       .
   JCL for Assembler
       .
       .
//SYSIN  DD   *
       .
       .
   Data for Assembly
       .
       .
/*
//STEP2  EXEC PGM=IEWL
       .
       .
   JCL for OS linkage editor
       .
       .
//SYSIN DD    *
       .
       .
   Control cards for linkage editor
       .
       .
/*
```

Figure 10. Stage 1 Output (Stage 2 Input) Job Stream

Chapter 5: The Loader Utility

Note: *The information in this chapter is preliminary and is to be used for planning purposes only.*

The Loader program is a utility by which the operating system can transfer a network control program load module from the host processor to the communications controller. The Loader can load several network control programs at once, each into a different controller.

The Loader has two modules. One is an OS/360 utility that may be invoked as any other OS/360 utility. The other module runs in the controller. At the time the Loader is invoked, the controller module is contained in a data area within the host processor Loader module. The host processor module loads the controller module into the controller via an initial program load (IPL) command.

Before the Loader utility loads the network control program into the controller, it loads a diagnostic routine, called the initial test routine. This routine tests the controller for hardware malfunctions that might later cause malfunctioning of the network control program. If the initial routine detects no malfunctions, the Loader loads the network control program into the controller. If the initial test routine detects trouble, that routine stops. The subsequent attempt by the host processor module to load the network control program will fail, the Loader will produce an error message indicating the fact, and the Loader will continue to load any remaining controllers, if any remain to be loaded.

Initial testing is optional, but is recommended because it can prevent later network control program failure.

The Loader utility executes the initial test unless you specifically indicate in the utility control statement (LOAD) that the testing is to be omitted.

Successful completion of the network control program loading process is indicated to the CPU operator by a Write-to-Operator message. A separate message is issued for each successfully loaded controller.

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).

Error messages are given in Appendix C.

Host Processor and Controller Requirements

The host processor module of the Loader executes in a minimum MFT partition (48K bytes) or MVT region (50K bytes).

A maximum of three 2314 tracks are required, plus one directory block on SYS1.LINKLIB.

No work data sets are required for Loader execution.

The controller module of the Loader can be executed in any communications controller. The only requirements for loader operation are that the controller be identified to the operating system under which the Loader utility is running, that it be free to be allocated to the Loader job step, and that its power be on.

Inputs 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 network control program load module to be loaded.
- A data set containing LOAD statements specifying the names of the network control program load modules and the controller into which each is to be loaded.
- A partitioned data set containing the initial test module to be loaded before network control program loading. This data set is optional; it may be omitted if the initial test is not desired (as indicated by DIAG=N in the LOAD statement).

Outputs from the Loader Program

The Loader produces one output data set, the message data set (SYSPRINT). This contains completion or error messages produced by the Loader.

Job Control Statements

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

```
//            JOB      (initiates the job)

//            EXEC     (specifies the program name, IFLOADRN,
                       or the name of a procedure containing
                       the job control statements)

//SYSPRINT    DD       (specifies a sequential data set; the
                       data set can be sent to the SYSOUT de-
                       vice, magnetic tape volume, or direct-
                       access volume)

//SYSUT1      DD       (specifies the DASD input data set
                       containing the network control program
                       load modules)

//SYSUT3      DD       (specifies the DASD input data set
                       containing the communications control-
                       ler initial test; not required if
                       DIAG=N is included in LOAD statement)

//ccname      DD       (specifies the unit address of the
                       communications  controller to be  load-
                       ed)

                       One DD statement is required for each
                       communications  controller to be load-
                       ed.

//SYSIN       DD       (specifies the data set [input stream]
                       containing  the utility control state-
                       ments, LOAD)
```

Utility Control Statement (LOAD)

There is one utility control statement for the Loader: the LOAD statement. It specifies which member of the input data set contains the network control 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 loading.

```
LOAD    LOADMOD=member name,
        3705=(ddname)[,DIAG={Y}]
                           {N}
```

`LOADMOD=member name`

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

`3705=(ddname)`

Specifies the ddname given to the DD statement identifying the communications controller to be loaded.

`[DIAG={Y}]`
`      {N}`

Specifies whether or not the Loader is to load the initial test into the communications controller before loading the network control program load module (DIAG=Y if yes, DIAG=N if no).

Example of Job and Utility Control Statements

Assume that a network control program load module named NCP1 residing on a data set named ALLNCPS is to be loaded into the controller whose unit address is 030.

The control and utility statements would be similar to:

```
//CCLOAD      JOB      123456,SMITH,MSGLEVEL=1
//            EXEC     PGM=IFLOADRN
//SYSPRINT    DD       SYSOUT=A
//SYSUT1      DD       DSNAME=ALLNCPS,UNIT=2311,          X
//                     VOL=SER=111111,DISP=OLD
//SYSUT3      DD       DSNAME=INITEST,UNIT=2311,          X
//                     VOL=SER=222222,DISP=OLD
//CC030       DD       UNIT=030
//SYSIN       DD       *
            LOAD       LOADMOD=NCP1,3705=CC030
```

This example assumes that the initial test routine is to be loaded and executed before the network control program is loaded. If the initial test is not wanted, the LOAD statement would also includes DIAG=N and the SYSUT3 DD statement is omitted.

Chapter 6: The Dump Utility

Note: *The information in this chapter is preliminary and is for planning purposes only.*

The Dump program lets you examine the contents of controller storage and local store registers as an aid to error diagnosis.

Either a complete or a partial dump may be requested, and the output listing may be either formatted or unformatted. Both types of listings show the hexadecimal representation of controller storage and register contents and give the character equivalents of all EBCDIC bit patterns that represent characters. The formatted listing shows, in addition, certain of the network control program control blocks. These control blocks appear at the beginning of the dump listing. Optionally, the listing can show the mnemonic operation codes.

The Dump program is made up of a module that runs in the host processor under the operating system and a module that is executed in the controller. The controller module is contained within the host processor module until the latter sends it, via an IPL 4 command, to the controller.

The Dump program transfers the entire contents of controller storage and local store registers to the host processor module, which places them on a direct access data set. When this is completed, the utility informs the CPU operator. At this point the controller is idle and can be reloaded with a network control program via the Loader utility.

The Dump program then analyzes the utility control statement, DUMP, and, as that statement directs, sends to an output data set all or a selected portion of the information received from the controller. Multiple DUMP statements may be used if listings of different parts of controller storage are required, or if multiple dumps are required for any reason.

If errors exist in the DUMP statement, the program sends the entire controller storage and register contents to the output data set.

The dump process destroys a small amount of the data in the controller storage, which therefore does not appear in the dump listing. The areas not available are:

- The first 512 bytes of storage (this is the read-only storage [ROS] bootstrap area)
- A 256-byte area from storage location (decimal) 1024 to 1280.

Note: *Only the local store registers appear in the dump listing. (If the contents of external registers must be examined, they must be displayed on the controller operator panel and the contents noted before the Dump program is executed.)*

Host Processor and Controller Requirements

The host processor module of the Dump program is executable on any System/360 or System/370 that will accommodate the System/360 Operating System (in either the MFT or MVT versions). The host processor module of the Dump program executes in a mimimum MFT partition (48 bytes).

A maximum of three IBM 2314 tracks are required, plus two directory blocks on SYS1.LINKLIB.

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 requires 2(32) + 1 = 65 512-byte blocks (eight 2314 tracks).

The controller module of the Dump program is executable in any communications controller. (There are two versions of the controller module; one for a controller equipped with a Type 1 channel adapter, the other, for a Type 2 channel adapter.)

Inputs to the Dump Program

The only input required for the Dump program is a DUMP control statement appearing in the input job stream (SYSIN).

Outputs from the Dump Program

The Dump program produces three outputs:

- A work data set to which the entire contents of controller storage and local store registers are sent.
- An output data set to which the formatted or unformatted dump listings are sent. The data set also contains certain error messages.
- The system message class data set. This contains completion or error messages produced by the Dump program.

Job Control Statements

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

```
//            JOB      (initiates the job)

//            EXEC     (specifies the first job step, IFL-
                       READ, or the name of a procedure con-
                       taining the job control statements).

//SYSUT1      DD       (specifies the communications control-
                       ler the contents of which are to be
                       dumped)

//SYSUT2      DD       (specifies the DASD work data set onto
                       which the contents of the communica-
                       tions controller are to be dumped)
```

```
//SYSPRINT    DD       (specifies a sequential data set
                       [system output device, magnetic tape,
                       or DASD volume] onto which IFLDUMP is
                       to place the dump listing)

//SYSIN       DD       (specifies the data set [input stream]
                       containing  the utility control state-
                       ment, DUMP)

/*
```

Utility Control Statement (DUMP)

The Dump program requires one control statement, DUMP. It specifies whether the dump listing will show the entire contents of communications controller storage or a portion thereof, whether the listing is to be formatted or unformatted, and whether the mnemonic operation codes are to appear in the listing. You may code a sequence of DUMP statements, if you wish more than one listing to be printed. For example, the low end of communications controller storage could be printed in one listing, the high end in another listing. The control statement format is:

```
DUMP    [FROMADDR=address]
        [,TOADDR=address]
        [,FORMAT={Y}]
                 {N}
        [,MNEMONIC= {Y}]
                    {N}
```

`[FROMADDR=address]`

Specifies the lower limit of the controller storage to appear on the listing. If FROMADDR is omitted, the listing will start at address X'200'.

`[TOADDR=address]`

Specifies the upper limit of controller storage to appear on the dump listing. If TOADDR is omitted, the listing will end at the upper limit of controller storage.

`[FORMAT={Y}]`
`        {N}`

Specifies whether or not the IFLDUMP program is to format the dump listing. The listing will be unformatted unless you specify FORMAT=Y.

`[MNEMONIC= {Y}]`
`           {N}`

Specifies whether or not the mnemonic operation codes are to be printed in the listing.

Example of Job and Utility Control Statements

Assume that a controller whose unit address is 030 is to be dumped, the dump listing is to show the contents of controller storage from address X'17F0' to the

end, the listing is to be formatted, and mnemonic operation codes are to be printed.

The control and utility statements would be similar to:

```
//CCDUMP      JOB       123456,SMITH,MSGLEVEL=1
//JOBLIB      DD        SYS1.DUMP,DISP=SHR,UNIT=2314,   X
//                      VOL=SER=333333
//EXEC        EXEC      PGM=IFLREAD
//SYSUT1      DD        UNIT=030
//SYSUT2      DD        UNIT=SYSDA,DISP=NEW,            X
//                      SPACE=(512,(513),,CONTIG),      X
//                      DCB=(DSORG=DA)
//SYSPRINT    DD        SYSOUT=A
//SYSIN       DD        *
            DUMP      FROMADDR=17F0,FORMAT=Y,MNEMONIC=Y
/*
```

Appendix A: Types of Stations Supported by IBM 3705

The IBM 3705 Communications Controller 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 1050 Data Communication System[1,2]
IBM 2740 Communications Terminal[1,2,3]
IBM 2741 Communications Terminal[1,2,3]
IBM 2770 Data Communications System[4,5]
IBM 2780 Data Transmission Terminal[4,5]
IBM 2972 General Banking Terminal System[4,5]
IBM 3270 Information Display System[4,5]
IBM 3735 Programmable Buffered Terminal[4,5]

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; AT & T 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.

Transmission Control Units:

IBM 2701 Data Adapter Unit[4,5]
IBM 2703 Transmission Control[4,5]
IBM 2715 Transmission Control[4] Model 2
IBM 3705 Communications Controller[4,5]

Computers:

IBM System/3[4,5]
IBM System/7 (supported as a 2740 Model 1)
IBM System/360 Model 20 (with BSC Adapter)[4,5]
IBM System/360 Model 25 (with Synchronous Data Adapter)[4,5]
IBM System/370 Model 135 (with Integrated Communications Adapter with Synchronous Data Adapter 2)[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)

Appendix B: Network Control Program Generation Messages

The 3705 Assembler program produces diagnostic error messages during expansion of the network control program generation macro instructions. The message identifier for each message is 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 network control program being generated. Generation of the network control 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 network control program. Generation of the stage 2 job stream is therefore terminated.

IFQ
Is the identifier for network control program generation messages.

nnn
Is the message serial number.

I
Indicates that the message is for information. No action by the operator is required.

-----text-----
Is the text of the message explaining the error condition.

```
IFQ001I aaa=bbb INVALID, ccc IS ASSUMED.
```

Explanation:
bbb is not a valid specification for operand aaa.

System Action:
The generation procedure assumes the default value ccc.

Programmer Response:

If the default value ccc is not acceptable, correct operand aaa and resubmit stage 1.

IFQ002I yyy INVALID NAME, EXCEEDS 8 CHARACTERS, {zzz IS ASSUMED. / IGNORED. / REQUIRED.}

Explanation:

The symbol yyy specified in the name field of the macro instruction exceeds 8 characters.

System Action:

The generation procedure takes one of these actions:

- zzz IS ASSUMED.

  The name zzz is assumed as the name of the macro. Generation continues.

- IGNORED.

  A name is not required and is ignored. Generation continues.

- REQUIRED.

  A name is required. Generation is terminated.

Programmer Response:

If the name is required or the name assumed is not acceptable, correct the name field of the macro and resubmit stage 1.

IFQ003I yyy INVALID NAME, FIRST CHARACTER NOT ALPHABETIC, {zzz IS ASSUMED. / IGNORED. / REQUIRED.}

Explanation:

The symbol yyy specified in the name field has a non-alphabetic first character.

System Action:

The generation procedures takes one of these actions:

- zzz IS ASSUMED.

  The name zzz is assumed as the name of the macro. Generation continues.

- IGNORED.

  A name is not required and is ignored. Generation continues.

- REQUIRED.

  A name is required. Generation is terminated.

Programmer Response:

If the name is required or the name assumed is not acceptable, correct the name field of the macro and resubmit stage 1.

IFQ004I NAME OMITTED, {yyy IS ASSUMED. / REQUIRED.}

Explanation:

No symbol was specified in the name field of the macro; a name is required.

System Action:

The generation procedure takes one of these actions:

- yyy IS ASSUMED.

  The symbol yyy is assumed as the name of the macro. Generation continues.

- REQUIRED.

  A name is required. Generation is terminated.

Programmer Response:

If the name is required or the name assumed is not acceptable, correct the name field of the macro and resubmit stage 1.

IFQ005I bbb-INVALID SUBOPERAND, IGNORED.

Explanation:

The value bbb is invalid for this suboperand.

System Action:

The invalid suboperand value is ignored. Generation continues.

Programmer Response:

If the suboperand is required and its value cannot be supplied during execution of the network control program (by a control request sent to the network control program or by network control program initialization), correct the suboperand by providing a valid suboperand and resubmit Stage 1.

```
IFQ006I SEQUENCE ERROR-mmm,... NOT DEFINED.
```

Explanation:

The macro or macros mmm do not appear in the network control 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).

System Action:

Generation is terminated.

Programmer Response:

Insert the macro or macros mmm in the correct sequence in the input statements and resubmit stage 1.

```
IFQ007I SEQUENCE ERROR-mmm1 PRECEDES mmm2.
```

Explanation:

Macro mmm2 does not precede macro mmm1 in the network control program generation input statements.

System Action:

Generation is terminated.

Programmer Response:

Correct the sequence of macros mmm1 and mmm2 and resubmit stage 1.

```
IFQ008I TERM=bbb, NON-SUPPORTED TERMINAL TYPE.
```

Explanation:

The type of terminal indicated by bbb is not a type supported by the network control program.

System Action:

Generation is terminated.

Programmer Response:

Correct the value bbb and resubmit stage 1.

```
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:

- ccc is ASSUMED.

  The default value ccc is assumed. Generation continues.

- REQUIRED.

  A value is required. Generation is terminated.

Programmer Response:
If a value is required or the value assumed is not acceptable, correct the value and resubmit stage 1.

```
IFQ010I NO VALID BH EXECUTION POINTS, ALL IS ASSUMED.
```

Explanation:
One or more of the block handler execution points were specified in the BHEXEC operand, but none were valid.

System Action:
The default value ALL is assumed. Generation continues.

Programmer Action:
If the value ALL is not acceptable, specify the desired value in the BHEXEC operand and resubmit stage 1.

```
IFQ011I bbb PREVIOUSLY SPECIFIED, IGNORED.
```

Explanation:
The suboperand value bbb was specified more than once for the operand being processed.

System Action:
The duplicate value is ignored. Generation continues.

Programmer Response:
If the omission of the duplicate value does not provide the correct value for the operand, specifiy the correct value or values in the operand and resubmit stage 1.

IFQ012I bbb INVALID, ccc ASSUMED.

Explanation:

The suboperand value bbb is not one of the valid values for the operand being processed.

System Action:

The default value ccc is assumed. Generation continues.

Programmer Response:

If the value ccc is not acceptable, specify the correct value in the operand and resubmit stage 1.

IFQ013I PARAMETERS CONFLICT, explanation.

Explanation:

One or more of the parameters specified conflict. The explanation defines the conflicting parameters.

System Action:

The explanation in the message describes the system action taken.

Programmer Response:

If the action described in the explanation part of the message is not acceptable, respecify the parameters so they do not conflict and resubmit stage 1.

IFQ014I bbb CPS-NON STANDARD.

Explanation:

The value bbb specified as one of the data rates for the CSB macro being processed is not one of the standard data rates for a communication scanner.

System Action:

The data rate bbb is accepted as valid. Generation continues.

Programmer Response:

If one of the standard data rates for a communication scanner was intended, correct the value bbb and resubmit stage 1. If the value indicated was intended, no action is required.

IFQ015I aaa INVALID ON CONTINUATION-IGNORED.

Explanation:

The operand aaa is specified on a continuation statement of the macro being processed. This operand, if specified, must appear on the first macro statement, not on a continuation statement.

System Action:

None. Generation continues.

Programmer Response:

If the value assigned for aaa on the first macro statement is the desired value, no programmer action is required. If the value assigned is not the desired value, specify a correct value on the first macro statement and resubmit stage 1.

IFQ017 BLOCK HANDLER NOT GENERATED.

Explanation:

The block handler specified is not generated because of errors in specifying the block handling routine function macros.

System Action:

The block handler is not generated. Generation of the network control program is terminated.

Programmer Response:

Correct the block handling routine function macros and resubmit stage 1.

IFQ018I PREVIOUS BLOCK HANDLER MAY BE INCOMPLETE.

Explanation:

The generation procedure encountered two STARTBH macros without an ENDBH macro between the two. The block handler defined by the first STARTBH macro may be incomplete (that is, one or more of the function macro statements may be missing from the network control program generation input statements).

System Action:

The block handler beginning with the first STARTBH macro is generated as though an ENDBH macro appeared in the input statements preceding the second STARTBH macro.

Programmer Response:

Examine the macro statements following the first STARTBH macro. If any statements are missing, supply the missing statements and resubmit stage 1. If no statements are missing, no action is required.

IFQ019I {PT1 / PT2 / PT3} =bbb-INVALID BH NAME.

Explanation:

bbb is an invalid assembler language symbol. bbb is specified as the name for a block handler to be executed at point 1, 2 or 3 (PT1, PT2, or PT3).

System Action:

Generation is terminated.

Programmer Response:

Correct the symbol following PT1=, PT2=, or PT3= and resubmit stage 1.

IFQ020I mmm INVALID FOR BH EXECUTED AT {PT1. / PT2. / PT3.}

Explanation:

The function macro mmm is specified in a block handler to be executed at the point indicated (PT1, PT2, or PT3). This macro may not be performed at the point specified.

System Action:

The function specified by mmm is not included in the block handler. Generation continues.

Programmer Response:

If the block handler generated is incorrect, correct it by specifying the desired function macros, then resubmit stage 1.

IFQ022I aaa=bbb INVALID, EXCEEDS 8 CHARACTERS.

Explanation:

The symbol bbb is specified as the name of a macro in operand aaa. bbb contains more than eight characters.

System Action:
Generation is terminated.

Programmer Response:
Specify a valid assembler language symbol of no more than eight characters and resubmit stage 1.

IFQ027I aaa NOT SPECIFIED, REQUIRED FOR explanation.

Explanation:
Operand aaa is not specified; this operand is required for the reason given in the explanation part of the message.

System Action:
Generation is terminated.

Programmer Response:
Supply the missing operand and resubmit stage 1.

IFQ028I aaa=bbb INVALID, EXCEEDS n CHARACTERS.

Explanation:
The value bbb is specified in operand aaa. The number of characters in bbb exceeds the maximum, n, allowed for the operand.

System Action:
If the severity code is 4, bbb is ignored, and generation continues. If the severity code is 8, generation is terminated.

Programmer Response:
If a value is required for operand aaa or if the severity code is 8, specify a correct value in operand aaa and resubmit stage 1.

IFQ029I aaa=bbb INVALID, FIRST CHARACTER NOT ALPHABETIC.

Explanation:
The symbol bbb is specified as the name of a macro in operand aaa. The first character of bbb is not alphabetic.

System Action:
Generation is terminated.

Programmer Response:
Specify a valid assembler language symbol as the value in operand aaa and resubmit stage 1.

IFQ030I aaa NOT SPECIFIED-REQUIRED.

Explanation:
Operand aaa, which is required, is not specified.

System Action:
Generation is terminated.

Programmer Response:
Supply the missing operand and resubmit stage 1.

IFQ031I aaa=bbb INVALID-REQUIRED.

Explanation:
The value bbb is not valid for operand aaa.

System Action:
Generation is terminated.

Programmer Response:
Specify a valid value for operand aaa and resubmit stage 1.

IFQ032I mmm HAS PREVIOUSLY BEEN ENTERED-IGNORED.

Explanation:
Macro mmm is specified more than once in the network control program generation input statements. This macro may be specified only once.

System Action:
The repeated macro is ignored. Generation continues.

Programmer Response:

- 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 1.

IFQ033I NO TERMINALS DEFINED IN THIS GENERATION.

Explanation:

No TERMINAL macros are included in the network control program generation input statements. TERMINAL macros are required.

System Action:

Generation is terminated.

Programmer Response:

Insert, in the network control program generation input statements, TERMINAL macros for the stations in the teleprocessing network. Then resubmit stage 1.

IFQ035I CSB MOD=bbb SPECIFIED, ALL LOWER CSB'S 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.

Programmer Response:

Insert in the network control 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. Then resubmit stage 1.

IFQ037I CSB NOT DEFINED FOR { BASE MODULE. / EXPANSION MODULE 1. / EXPANSION MODULE 2. / EXPANSION MODULE 3. }

Explanation:

No CSB macro for the module indicated appears in the network control program generation input statements.

System Action:

Generation is terminated.

Programmer Response:

Insert, in the network control program generation input statements, a CSB macro for the indicated module, then resubmit stage 1.

Appendix C: 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 and dump programs.

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 IFL010 is also sent). If the DD statement is missing, another system message (IEC130I) will also be issued.

Utility Action:

The loading process is terminated.

Programmer Response:

Check the DD statement indicated by ddname for correct specification and check that the 3705= parameter specifies the proper ddname.

```
IFL001I UTILITY END  xx WAS THE HIGHEST SEVERITY CODE
```

Explanation:

The loader utility has processed all control cards in the input data set. The return codes possible are:

00 The loading process was completed successfully; all 3705s that were to be loaded are now loaded.

04 The loading process for at least one of the 3705s to be loaded generated a warning message.

08 The loading process for at least one of the 3705s to be loaded was not successfully completed; the loading process generated an error message.

12 Because of a severe error, none of the 3705s to be loaded was successfully loaded.

Utility Action:

The loader job is completed.

Programmer Response:

If the severity code is greater than 0, examine the message data set for the appropriate messages. Correct the job control statements in error and resubmit the job. (The resubmitted job need not specify for loading, those 3705s, if any, that were successfully loaded.)

IFL002I ERROR-LOADING PROCESS TERMINATED-** LOADMOD RECORD SIZE TOO LARGE ***

Explanation:

The input record size of the network control program load module was too large for the buffer space available in the host processor (message IFL010 is also sent).

Utility Action:

The 3705 to be loaded with the indicated load module is not loaded; the loader utility processes the next utility control card, if any.

Programmer Response:

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 network control 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.

Utility Action:

The 3705 to be loaded with the indicated load module is not loaded; the loader utility processes the next utility control card, if any.

Programmer Response:

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 3705 during loading (message IFL010 is also sent.)

Utility Action:

Loading of the 3705 is terminated; the loader processes the next utility control card, if any.

Programmer Response:

Resubmit the loader job.

IFL005I ERROR-LOADING PROCESS TERMINATED ** INITIAL TEST DETECTED 3705 ERROR ***

Explanation:

The initial test routine did not return control to the loader utility. This indicates that a hardware error occurred in the 3705 that would prevent the network control program from executing properly (message IFL010 is also sent).

Utility Action:

The 3705 is not loaded; the loader utility processes the next utility control card.

Operator Response:

The operator should follow the problem determination procedure.

IFL006I ERROR-LOADING PROCESS TERMINATED ** CONTROL STATEMENT ERROR ***

Explanation:

The LOAD utility control card contained a syntax error.

Utility Action:

The 3705 is not loaded; the loader utility processes the next utility control card.

Programmer or Operator Response:

Correct the erroneous LOAD card and resubmit the loader job.

IFL007I ERROR-LOADING PROCESS TERMINATED ** PROGRAM FAILURE IN 3705 ***

Explanation:

The 3705 module of the loader utility encountered a software or hardware error in the 3705 (message IFL010 is also sent).

Utility Action:

Loading of the 3705 is terminated; the loader utility processes the next utility control card.

Operator Response:

The operator should follow the problem determination procedure.

IFL008I 3705 LOAD COMPLETE 3705=xxx LOADMOD=(member)

Explanation:

The 3705 whose unit address is xxx was successfully loaded with the network control program load module whose member name is specified by (member). (This message is sent to the programmer and the operator).

Utility Action:

The loader utility processes the next control card.

Programmer or Operator Response:

None.

IFL009I WARNING-LOADING PROCESS COMPLETED ** LOAD MODULE LARGER THAN 3705 ***

Explanation:

The load module is too large for the 3705. The loader utility loaded as much of the load module as possible in the 3705 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 3705 specified by the 3705= parameter, or the network control program specified is too large for the 3705 and must be reduced in size.

Utility Action:

The loader utility processes the next utility control card.

Programmer Response:

Correct the LOAD utility control card or regenerate a network control program of a size that the 3705 can accommodate.

IFL010I 3705 LOAD FAILED 3705-xxx LOADMOD=(member)

Explanation:

The loading of the 3705 indicated by xxx 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.

Operator 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.

Utility Action:

The loader utility processes the next LOAD utility control card.

Programmer Response:

Correct the erroneous LOAD card and resubmit the loader job.

IFL012D 3705-xxx ACTIVE *** REPLY TO CONTINUE

Explanation:

An attempt has been made to load the xxx 3705, which contains an active control program.

Utility Action:

The loader utility waits on the operator's reply.

Operator Response:

If the 3705 should be loaded, enter REPLY xx, 'U'.This causes the loader utility to continue the load for this 3705. If the 3705 should not be loaded, enter REPLY xx, 'M' to terminate the load request; processing then continues with the next request.

IFL013I 3705 ACTIVE—LOAD CANCELLED BY THE OPERATOR

Explanation

The 3705 was in an active state and the operator did not want to continue the load.

Utility Action:

The loader utility continues with the next request.

Programmer or Operator Response:

None.

Messages Issued by the Dump Program

IFL100I xxxxxxxx NOT OPENED

Explanation:
The named data set (SYSUT1 or SYSUT2) could not be opened. Either the DD statement defining the data set was missing from the input job stream or a DCB parameter is invalid.

Utility Action:
The job step is terminated.

Programmer Response:
Ensure that a DD statement for the indicated data set appears in the input job stream and that the DCB parameter is correct. Then resubmit the job.

IFL102I THE 3705 COULD NOT BE DUMPED

Explanation:
A permanent I/O error occurred during processing of the SYSUT1 data set.

Utility Action:
The Dump job is terminated.

Programmer Response:
Ensure that the data set characteristics specified for the SYSUT1 data set are correct. If they are not, correct them and resubmit the job.

IFL103I THE 3705 HAS BEEN DUMPED SUCCESSFULLY

Explanation:
The contents of the 3705 have been successfully dumped to the work data set (SYSUT2).

Utility Action:
None.

Programmer Response:
None.

```
IFL104I D ddd,jjj,ppp,sss
```

Explanation:

D indicates that a request was entered to dump the contents of the 3705 whose address is ddd by job step sss of job jjj. If the IFLREAD job step was invoked by a cataloged procedure, the procedure step name was ppp.

Utility Action:

The Dump job waits for an operator response before dumping the 3705.

Operator Response:

If the request to dump the contents of the 3705 is to be honored, enter REPLY xx,'U'.

If the request is not to be honored, terminate the job step and notify the programmer.

```
IFL200I xxxxxxx NOT OPENED
```

Explanation:

The data set indicated by xxxxxxx (SYSUT2, SYSPRINT, or SYSIN) could not be opened. Either the DD statement defining the data set was not included in the input job stream, or a DCB parameter is invalid.

Utility Action:

The Dump job is terminated.

Programmer Response:

Probable user error. Ensure that a DD statement for the indicated data set is included in the input job stream and that the DD statement parameters are correct. Resubmit the Dump job.

```
IFL201I INVALID CONTROL STATEMENT; DEFAULT TAKEN
```

Explanation:

The DUMP statement contains an error.

Utility Action:

The Dump utility provides an unformatted dump of the entire contents of 3705 storage.

Programmer Response:

Probable user error. If the unformatted dump provided does not provide sufficient information, correct the DUMP statement and resubmit the Dump job.

IFL202I UNABLE TO PRODUCE REQUESTED DUMP

Explanation:

A permanent I/O error prevented the Dump utility from formatting and sending to the SYSPRINT data set the contents of the 3705 storage.

Utility Action:

The Dump job is terminated.

Programmer Response:

Resubmit the Dump job.

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 network control program in the communications controller.

Addressing. The means whereby the originator or control unit selects the teleprocessing device to which it is going to send a message.

Address trace. A service aid by which the contents of selected areas of communications controller storage, and selected external registers, can be recorded at each successive interrupt.

Alternate path retry. An optional facility of the network control program that allows the user to specify for certain device types a line to be used as a backup line if the primary line becomes unavailable due to an irrecoverable error.

Block. The smallest data unit recognized by the 3705. For start-stop devices, a unit of data between two EOB characters; for BSC devices, a unit between an STX or SOH character and an ETB or ETX character.

Block handler (BH). A group of block handling routines that are executed sequentially to process a block of data at a specified point in its path through the network control program.

Block handler (BH) set. A group of block handlers. A BH set may be associated with one or more teleprocessing devices.

Block handling macro (BH macro). One of the network control program generation macros that describe optional block processing functions to be included in the network control program.

Block handling routine (BHR). A routine that performs a single processing function for a block of data passing through the network control program. A typical BHR function is inserting the date and time of day in the block.

Buffer. A temporary storage area for data.

Buffer pad characters. A sequence of characters that the network control program sends to an access method buffer preceding message data, to allow space for the access method to insert message prefixes.

Channel adapter (CA). A 3705 hardware unit that provides attachment of the 3705 to a System/360 or System/370 channel.

Checkpoint/restart. A facility that allows a program to return to a previous point and resume execution there on the basis of information stored at that point when execution was suspended.

Cluster. A station that consists of a control unit and the terminals attached to it. In this publication, a *cluster-type* station refers to an IBM 2972 General Banking Terminal System or an IBM 3270 Information Display System.

Communication scanner. A 3705 hardware unit that provides the interface between line interface bases and the central controller. The communication scanner monitors the communication lines for service requests.

Component. An independently addressable part of a station that performs either an input or an output function for the terminal, but not both.

Conditional operand. An operand of a network control program generation macro instruction that must be coded or omitted depending on whether certain other operands are coded or omitted.

Configuration macro. One of the network control program generation macros that provide information necessary to construct the tables needed by the network control program to control the flow of data between the 3705 and stations, and between the 3705 and the host processor.

Device. (See Teleprocessing device.)

Dial set. A user-specified combination of switched point-to-point lines from which the network control program selects a line with which to communicate with a station.

Dynamic. Occurring at the time a program is executed.

Dynamic buffering. Allocating storage as it is needed for incoming data during program execution.

Dynamic control function. One of the network control program functions initiated by a Control request from the teleprocessing access method.

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.

Element. A part of the teleprocessing network defined by a network control program generation macro. Possible elements are line groups, lines, clusters, terminals, and components.

Error recovery procedure (ERP). A program that auto-

matically attempts to correct a transmission error.

Formatted dump. A dump in which certain network control program control blocks are isolated and identified.

Generation delimiter macro. The network control program generation macro that marks the end of the network control program generation input stream.

Host processor. The central processing unit to which the 3705 is attached by a channel and that executes the teleprocessing access method that supports the 3705.

Initial status. The status of a communication line or teleprocessing device at the moment the network control program is loaded into the communications controller and given control.

Initial test routine. A diagnostic program executed in the 3705 before the network control program is loaded. The initial test routine tests the 3705 hardware for conditions that might cause failure after operation begins.

Interrupt priority. The order in which the network control program processes interrupts received simultaneously from two or more communication lines.

Invite. A network control program teleprocessing command that starts a session with a teleprocessing device by allowing the device to send data to the host processor.

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 3705.

Line interface base (LIB). A 3705 hardware unit that provides for the attachment of up to 16 communication lines to the 3705.

Load module. A program in a format suitable for loading into storage for execution. A *Network Control Program* load module is produced by the linkage editor during the network control program generation procedure; the Loader utility loads it into the 3705.

Loader program. A utility program operating partly in the host processor and partly in the communications controller, that transfers a network control program load module from host processor storage to the communications controller.

Logical line group. A user-specified combination of lines that do not necessarily have the same characteristics. A logical line group allows the access method to refer, in a request, to several lines collectively instead of individually.

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.

Network Control Program generation language. The set of macro instructions and associated operands by which the 3705 user defines the network configuration and operating parameters of the teleprocessing subsystem.

Network Control Program generation procedure. A two-stage process that creates a Network Control Program load module based on parameters specified by the user through the network control program generation language.

Network Control Program. The IBM-supplied control program for the 3705.

Online terminal testing. A diagnostic aid by which a terminal or console may request any of several kinds of tests to be performed upon either the same terminal or console or a different one.

Parameter. A variable that is given a constant value for a specific purpose or process.

Pause-retry. A network control program option that allows the user to specify how many times the network control program should try to retransmit data after a transmission error occurs, and how long the network control program should wait between each attempt.

Polling. A technique by which each of the teleprocessing devices sharing a communication line is interrogated to determine whether it has data to send.

Program check. An error in a program that suspends execution.

Record. A group of related data items treated as a unit.

Request. A directive from the access method that causes the network control program to perform a data transfer operation or auxiliary operation.

Reset. A network control program Teleprocessing command that stops the current operation for a teleprocessing device and disposes of any requests still outstanding for that device.

Resource. Any facility of a computing system or operating system required by a job or task, including main storage, input/output devices, processing time, etc.

Response. The information the network control program

sends to the access method, usually in answer to a request received from the access method. (Some responses, however, result from conditions occurring within the network control program, such as accumulation of error statistics.)

Restart command. A network control program command that requests that the 3705 be started using the most recent checkpoint records.

Service order table. The list of teleprocessing devices on a multipoint line (or point-to-point line where the terminal has multiple components) in the order in which they are to be serviced by the network control program.

Service seeking. The process by which the network control program interrogates teleprocessing devices on a multipoint line (or point-to-point line where the terminal has multiple components) for requests to send data or for readiness to receive data.

Service-seeking pause. A user-specified interval between successive attempts at service seeking on a line when all teleprocessing devices on the line are responding negatively to polling.

Session. A series of command and data interchanges between the host processor and a teleprocessing device.

Session limit. The maximum number of concurrent sessions that can be initiated on a multipoint line (or point-to-point line where the terminal has multiple components).

Station. A point in a teleprocessing network at which data can either enter or leave. In this publication, a station refers to any of the computers, transmission control units, and terminals in the teleprocessing network connected to the 3705.

System designer. The individual who determines the teleprocessing equipment, network configuration, and communication services that constitute a teleprocessing subsystem.

System macro. One of the network control program generation macros that provide information pertaining to the entire 3705.

Teleprocessing. A form of information handling in which a data processing system utilizes communication facilities.

Teleprocessing command. One of the network control program commands that control the activity on the communication lines.

Teleprocessing device. A unit of teleprocessing equipment connected to the 3705 via a communication line and identified as a cluster, terminal, or component at network control program generation time.

Teleprocessing network. The stations that are controlled by a single access method (or, in the 3705, by a single network control program), and the communication lines by which they are connected to the transmission control unit.

Teleprocessing subsystem. The part of a data processing system devoted to the transfer of data across communication lines. The subsystem consists of the stations, data sets (or modems), communication lines, and the transmission control unit.

Terminal. A teleprocessing device capable of transmitting or receiving data (or both) over a communication line.

Test command. A network control program command that indicates that an on-line terminal test is to be performed for the specified teleprocessing device.

Test request message. A message entered from a terminal or console requesting that a specified online terminal test be performed upon that terminal or console or a different one. The network control program passes the test request message to the teleprocessing access method.

Trace table. An area within the network control program into which address trace information is placed.

Transmission. For start-stop devices, the data unit between a © and a ® line control character; for BSC devices, the data unit between an SOH or STX character and an EOT character.

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 a single channel to the computer.

Transmission limit. The maximum number of transmissions that can be sent to or received from a teleprocessing device during one session on a multipoint line (or point-to-point line where the terminal has multiple components) before the network control program suspends the session to service other devices on the line.

Unit of data transfer. One of the logical entities in which the network control program sends data to or receives data from stations in the Teleprocessing Network. The three units are the block, the message, and the transmission.

User block handling routine. A block handling routine coded by the user and added to the program network control program during program generation.

Index

YOUR COMMENTS, PLEASE . . .

Your answers to the questions on the back of this form, together with your comments, help us produce better publications for your use. Each reply is carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in using your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

Fold Fold

FIRST CLASS
PERMIT NO. 569
RESEARCH TRIANGLE PARK
NORTH CAROLINA

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY . . .

IBM Corporation
P. O. Box 12275
Research Triangle Park
North Carolina 27709

Attention: Publications Center, Dept. E01

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 Along Line

3705 Network Control Program Generation and Utilities, Guide and Reference Manual Printed in U.S.A. GC30-3000-0

READER'S COMMENT FORM

IBM 3705 Communications Controller
Network Control Program Generation and Utilities,
Guide and Reference Manual

Order No. GC30-3000-0

- How did you use this publication?

  As a reference source ☐
  As a classroom text ☐
  As ..... ☐

- Based on your own experience, rate this publication . . .

As a reference source:	.....	....	.....	.....	.....
	Very Good	Good	Fair	Poor	Very Poor
As a text:	.....	....	.....	.....	.....
	Very Good	Good	Fair	Poor	Very Poor

- What is your occupation? .....

- We would appreciate your other comments; please give specific page and line references where appropriate. If you wish a reply, be sure to include your name and address.

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.